

AD-A048 554

ARMY CONCEPTS ANALYSIS AGENCY BETHESDA MD
COMPUTER ASSISTED MATCH PROGRAM (CAMP), (U)
AUG 76 G L MARTIN, E R MONTAGNE
CAA-D-76-5

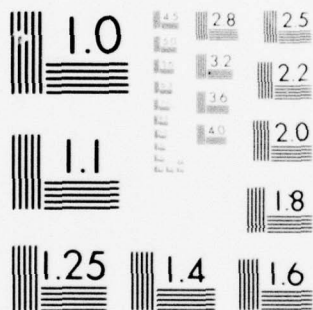
F/O 15/7

UNCLASSIFIED

NL

1 OF 4
AD-A048554





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AD-A048554

COMPUTER ASSISTED SEARCH PROGRAM

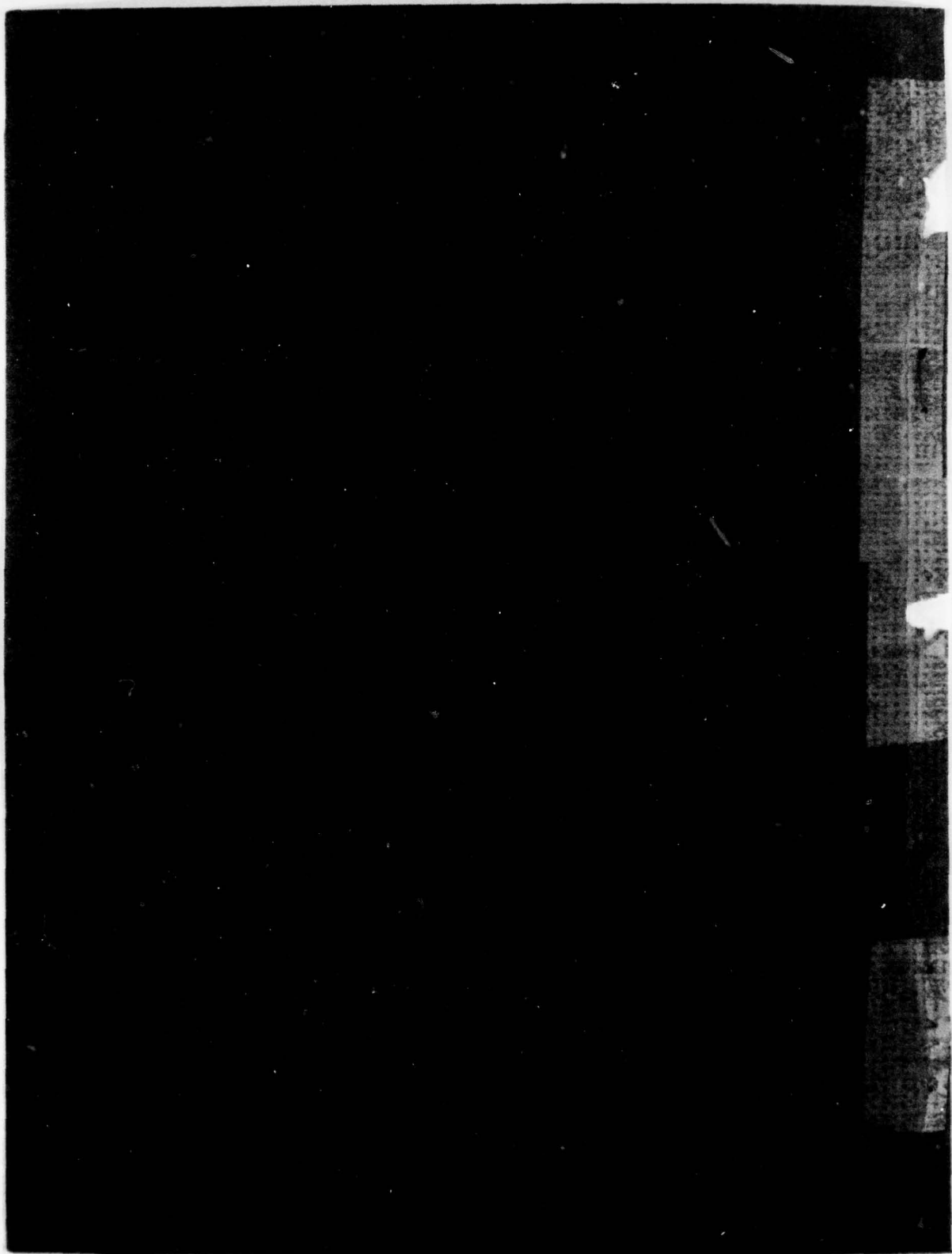
(CONT)

AD-A

D D
REC
JAN 2
R
1

DISTRIBUTION STATEMENT A

Approved for public release;
Distribution Unlimited



①

DOCUMENTATION

CAA-D-76-5

COMPUTER ASSISTED MATCH PROGRAM

(CAMP)

AUGUST 1976

AD-A

PREPARED BY

JOINT AND STRATEGIC FORCES DIRECTORATE

US ARMY CONCEPTS ANALYSIS AGENCY

8120 Woodmont Avenue

Bethesda, Maryland 20014

ADDITIONAL TO	
DTIC	White Section <input checked="" type="checkbox"/>
DDC	Ref Section <input type="checkbox"/>
UNANNOUNCED	<input type="checkbox"/>
JUSTIFICATION	
Per DDC Form 50	
BY on file	
DISTRIBUTION/AVAILABILITY CODES	
Dist.	AVAIL. and/or SPECIAL
A	

DDC
RECEIVED
JAN 23 1978
D

DISTRIBUTION STATEMENT A

Approved for public release;
Distribution Unlimited

UNCLASSIFIED
UNCLASSIFIED

THIS PAGE (When Data Entered)

DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER CAA-D-76-5	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Computer Assisted Match Program (CAMP)		5. TYPE OF REPORT & PERIOD COVERED Model Documentation
		6. PERFORMING ORG. REPORT NUMBER CAA-D-76-5
7. AUTHOR(s) Mr Gary L Martin MAJ Ernest R Montagne Jr		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS US Army Concepts Analysis Agency Joint and Strategic Forces Directorate 8120 Woodmont Avenue, Bethesda, MD 20014		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS US Army Concepts Analysis Agency Joint and Strategic Forces Directorate 8120 Woodmont Avenue, Bethesda, MD 20014		12. REPORT DATE 13 August 1976
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) NA		13. NUMBER OF PAGES 314
		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE NA
16. DISTRIBUTION STATEMENT (of this Report) <div style="border: 1px solid black; padding: 5px; text-align: center;">DISTRIBUTION STATEMENT A Approved for public release; Distribution Unlimited</div>		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Force structure, Force match, Movement requirements, Strategic mobility		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) CAMP is a data processing system designed to be used as a tool in force structuring and analysis and strategic mobility analysis. CAMP contains two major functions: Force match algorithm (FMA) and Movement Requirements Generator. CAMP interfaces DA force planning files (Force Accounting System), the FASTALS theater roundout model, and various logistical data files to produce movement requirements in the Mobility Requirements for Staff Analysis system (MORSA) and/or SMOBSMOD format.		

UNCLASSIFIED

UNCLASSIFIED

DECLASSIFICATION AUTHORITY
APPROVED FOR PUBLIC RELEASE
UNLIMITED RELEASE

UNCLASSIFIED

UNCLASSIFIED

TABLE OF CONTENTS

	Page
Disclaimer	ii
Report Documentation Page, DD Form 1473	iii
Table of Contents	vii
List of Figures	ix
List of Tables	ix
CHAPTER	
I System Introduction	I-1
Purpose	
Background	
General System Description	
Operation and Maintenance	
Security	
Limitations and Constraints	
II Functional Description - Force Match Algorithm (FMA)	II-1
Introduction	
System Processing Flow	
Interfaces with other models	
User Constraints	
III Functional Description - Movement Requirements Generator (MRG).	III-1
Introduction	
System Processing Flow	
Interfaces with other models	
User Constraints	
IV Data Dictionary	IV-1
V Job Preparation	V-1
FMA Job Preparation	
MRG Job Preparation	

VI Job Utilization VI-1

Force Match Algorithm
Movement Requirements Generator

VII Program Listings VII-1

	Listing Page Number
BUILD	1
SORTUDS	4
FSORT	12
RSORT	30
MATCH	34
ATL	45
ALT	49
LAYIN	51
BINCOPY	58
SORTARLOC	64
MORSAROLL	69
TUCHAROLL	79
POMCUSROLL	89
GEO	99
LOG	106
UPMDF1	113
NUR	115
MERGEMORSA	125
PCKAGE	126
INLAY	133
PRTPCK	138
BUILDMRG	140

VIII Sample Output VIII-1

APPENDICES

A. Study Contributors	A-3
B. References	B-3
C. Glossary	C-3

LIST OF FIGURES

Figure		Page
I-1	CAMP Input-Processing-Output Chart	I-2
II-1	FMA Input-Processing-Output Chart	II-4
III-1	MRG Input-Processing-Output Chart	III-3

LIST OF TABLES

II-1	FMA Programs	II-3
III-1	MRG Programs	III-2
V-1	FAS File Record	V-2
V-2	CAMP SUM File Record	V-3
V-3	RSORT Card Type 1	V-5
V-4	RSORT Card Type 2 (Lockout UIN)	V-5
V-5	Hand Played UIC Data	V-5
V-6	Lockout ADC01 Data	V-6
V-7	Alternate Theater ADC01 Data	V-6
V-8	MCODE 1 (POMCUS) ROBCO Data	V-7
V-9	MCODE 0 (In-country) Data	V-7
V-10	Lockout COMPO Data	V-7
V-11	Subperiod RDD Data	V-9
V-12	Notional Unit Data	V-9
V-13	Notional Unit PECOD Data	V-10
V-14	Above-the-Line RDD Data	V-10
V-15	ARLOC File Record Description	V-10
V-16	TUCHA ABF1 Record Description	V-11
V-17	TUCHA F2 Record Description	V-11
V-18	POMCUS F1 Record Description	V-12
V-19	POMCUS F2 Record Description	V-12
V-20	POMCUS F3 Record Description	V-12
V-21	MORSA File Record Description	V-14
V-22	MORSAROLL Input Data	V-15
V-23	GEO Data	V-16
V-24	Node Table	V-16
V-25	LOG Data	V-17
V-26	UPMDF1 Data	V-17
V-27	NUR Data	V-18
V-28	PCKAGE Data	V-20
V-29	INLAY Data	V-21
VI-1	SORTUDS Report	VI-2
VI-2	FSORT Reports	VI-3
VI-3	RSORT Report	VI-4

VI-4	MATCH Report	VI-5
VI-5	Data Elements Displayed by TUCAROLL	VI-7
VI-6	SMOBSMOD Cargo Categories	VI-10
VI-7	MDF1 Data Elements	VI-10
VI-8	Supply Status Summary	VI-11
VI-9	MRG Data Items Overlaid on FAS File	VI-13

COMPUTER ASSISTED MATCH PROGRAM (CAMP)

CHAPTER I SYSTEM INTRODUCTION

1. Purpose. The purpose of this documentation is to describe the Computer Assisted Match program (CAMP) Model in sufficient detail to allow use of CAMP by interested personnel.

2. Background

a. Various strategic mobility models require the input of force movement requirements in varying levels of detail.

b. Several studies conducted at the Concepts Analysis Agency (CAA) require production of input for the Strategic Mobility Simulation Model (SMOBSMOD), used at CAA, or the Mobility Requirements for Staff Analysis (MORSA) system, used at Joint Chiefs of Staff level. Automated production of such input is dictated by the volume of data required (typically, to 20,000 cards for MORSA). The CAMP was developed to meet this requirement.

3. General System Description. The CAMP is a data processing system developed at CAA. The system consists of two major functions as shown in Figure I-1: Force Match Algorithm (FMA) and Movement Requirements Generator (MRG). Each function includes a series of programs, primarily written in FORTRAN, but with some COBOL programs as well. The objectives of CAMP are to compare an actual Army force structure with time phased unit requirements for a given scenario, identify overages and shortfalls on the force, create notional units for shortfalls, and produce movement requirements (e.g., origin destination, required delivery date, and deployment weight) as determined from the time phased unit requirements for input to various strategic mobility models. The CAMP provides the interface between the Force Accounting System (FAS) used in force structuring and analysis by the DA staff, the Force Analysis Simulation of Theater Administration and Logistics Support (FASTALS) Model used at CAA, and the SMOBSMOD and MORSA mobility models. Significant features of CAMP are its abilities to generate notional FAS records for required units not on the FAS file; to develop unit movement requirements for the notional units and for actual units which are required by the scenario; and to calculate nonunit movement requirements (resupply, replacements, and fillers) to support the force. The user may specify up to seven theaters, each with different force lists, and nonunit factors. In general, CAMP programs function sequentially. Should an error in input

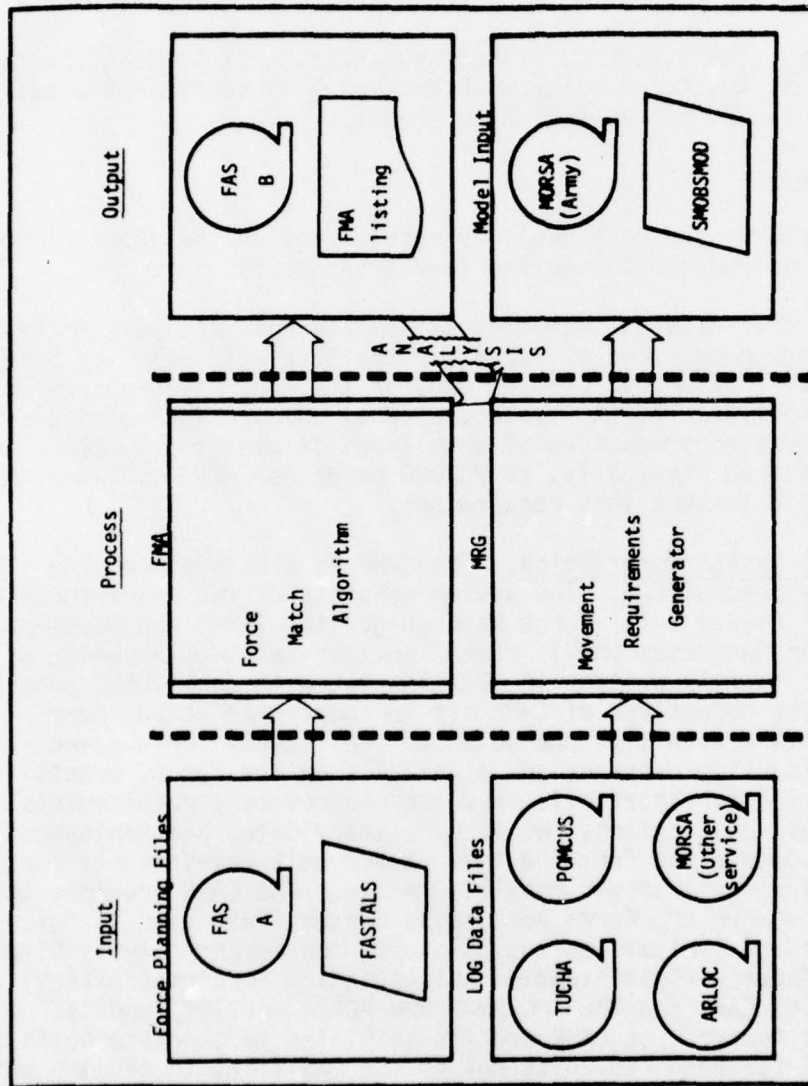


FIGURE I-1, CAMP Input-Processing-Output Chart

to one program lead to a processing error, execution is terminated and a diagnostic error message is printed to assist the user in correcting the error. The CAMP has been interfaced with the Unit Data System (UDS) to provide a generalized report generator capability.

4. Operation and Maintenance. The CAMP operates on the UNIVAC 1108 computer. The maximum program size is 58K words but most programs are between 20K and 40K words. Processing time is directly dependent on the size of the force being processed. Typically, a force may include on the order of 10,000 units, in which case exercise of the FMA will require approximately 1 hour of wall-clock time, and MRG will require approximately 2 hours. The CAMP is maintained by the CAA Methodology and Resources Directorate (MRD) that provides program-related user assistance. The proponent of CAMP is the Joint and Strategic Forces Directorate (JF), that provides functionally oriented user assistance.

5. Security. The CAMP programs are unclassified, but files and data are generally classified SECRET.

6. Limitations and Constraints

a. Data Base. Many of the files accessed by CAMP are not produced at CAA. The user often has little control over the quality of the data in these files. Close coordination with the data source is required to obtain valid data.

b. Formats. The input force must be in FAS file format. The output is produced only in SMOBSMOD and MORSA formats. Requirements for other formats should be coordinated with MRD.

COMPUTER ASSISTED MATCH PROGRAM (CAMP)

CHAPTER II FUNCTIONAL DESCRIPTION - FORCE MATCH ALGORITHM (FMA)

1. Introduction

a. The Force Match Algorithm (FMA) is a force structuring and analysis tool. The FMA compares an actual or planning force in FAS file format with a time phased requirements list produced by FASTALS. Selected units on the force may be locked out (excluded from consideration in the match) based on user input. For example, units located in a specific theater might be locked out in a match of requirements for another theater. The remaining units are match candidates and are matched against type unit requirements identified by FASTALS. Each match candidate is either matched or designated as excess to the stated requirements. Each requirement that cannot be matched with a unit on the force results in generation of a notional unit. Each matched unit and each notional unit are assigned a required delivery date (RDD) based on the FASTALS time phasing. Assignment of RDD within a FASTALS time period is user adjustable.

b. Output of the FMA is used by the force planner to allocate resources between active and reserve components. The rationale for allocation is that units required early should be active, while those required later may be reserve. Close scrutiny of overages and shortfalls is required to determine if substitutions can be made. It is from this manual analysis that CAMP draws the computer assisted portion of its name. The match cannot be fully automated because the force planner should have the final decision in structuring a force. However, CAMP output assists the planner in force analysis. In general, analysis by the planner will result in retention of many units designated as excess. Further, budgetary or other constraints may well result in the inability of the planner to activate the notional unit shortfalls. Only the application of sound judgment by the planner will result in designation of the appropriate force.

c. The FMA input consists of two basic files: the Force file and the Requirements file. The Force file is a planning force and may be considered as the starting point in the force structuring process. Typically, the Force file is provided in FAS format by the Office, Deputy Chief of Staff for Operations and Plans (ODCSOPS). The second input file is the Requirements file. This file is generated by FASTALS, the theater roundout model, in response to a specific scenario or situation.

2. System Processing Flow. The FMA consists of a series of nine programs that are identified in Table II-1.

a. Preprocessors

(1) Program BUILD accepts a FAS tape and creates a keyed access UDS file. The UDS allows detailed analysis of the force prior to match. In particular, the user can verify the number and location of various type units, examine the POMCUS packages, and update the file to correct errors or add new data. Capabilities of UDS are described in Reference 1.

(2) Program SORTUDS reads the UDS file. Records are sorted and written to a Force file (FF) on mass storage.

b. Main Programs. Input-processing-output charts for the main programs are shown in Figure II-1.

(1) Program FSORT reads the Force file to select Match Candidates. The user has considerable influence over the selection process. Certain fields of the FAS records can be updated based on user input. Subsequent to update, FSORT divides the force into five groups.

(a) Locked Out. These units are designated by the user based on the value of the ADCOI field. Locked Out units are excluded from consideration in the match process and are written to the Lock Out file (LOU). Generally, these will be units with specific missions in theaters not of interest in the current run. They are considered nondeployable.

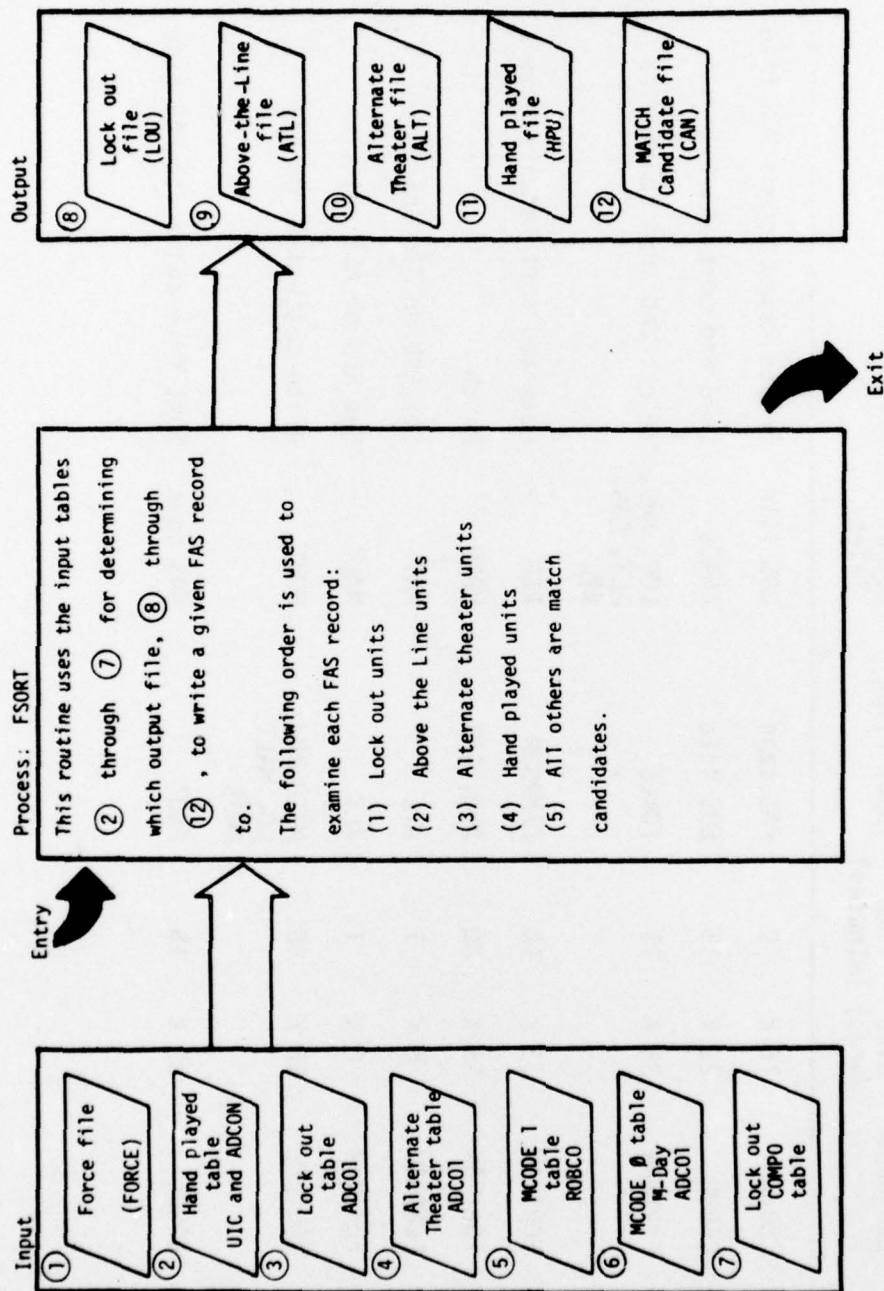
(b) Alternate Theater. These units are also designated by the user based on the ADCOI field. Alternate Theater units are required to be deployed to theaters other than the primary theater. Requirements for these units are not generated by FASTALS. The units are excluded from match consideration but will ultimately be assigned RDD and destination in the specified theater; these units are written to the Alternate Theater file (ALT).

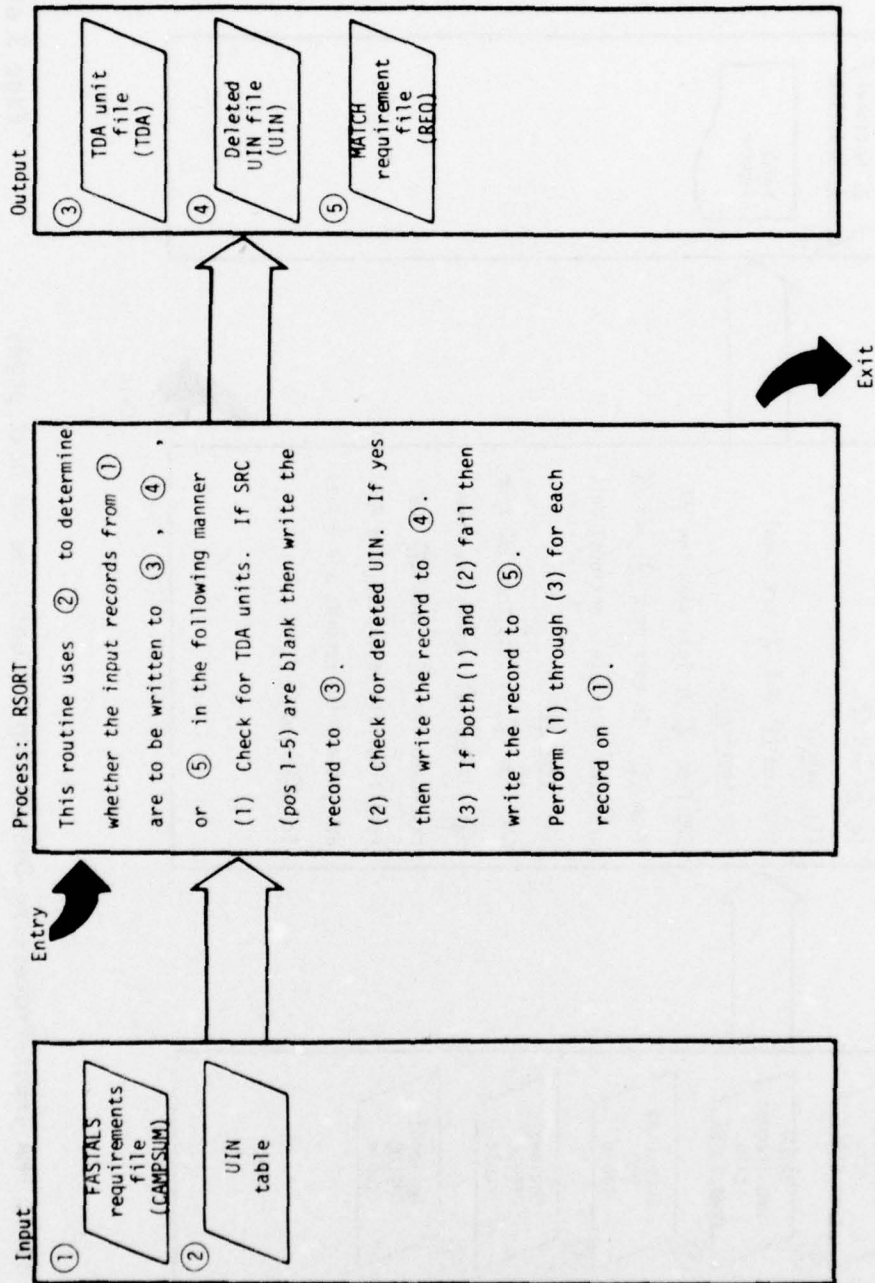
(c) Above-the Line. These are major combat forces (divisions and separate brigade sized units). Both the RDD and theater of these units are determined by the user. The Above-the-Line units are matched manually and are excluded from automated matching. These units are written to the Above-the-Line file (ATL).

(d) Hand Played Units. The user may specify a number of units (by UIC, ROBCO, or ADCOI) to receive specific RDD. Hand Played Units will be excluded from match processing but will be

TABLE II-1, Force Match Algorithm Programs

Program	Language	Size (words)	Time (minutes)	Input files	Output files	Function
BUILD	COBOL	16 K	2	FAS tape	UDS file	Create keyed access UDS file
SORTUDS	COBOL	28 K	15	UDS file	FORCE	Load and sort FAS file
FSORT	FORTAN	36 K	11	FORCE	LOU, HPU, ALT, CAN, ATL	Select and sort candidates
RSORT	FORTAN	33 K	10	CAMPSUM	REQ	Load and sort requirements
MATCH	FORTAN	35 K	30	REQ, CAN	MREQ	Match
ATL	FORTAN	8 K	1	ATL	MATL	Set RDD on ATL units
ALT	FORTAN	8 K	1	ALT	MALT	Set RDD on ALT units
L AYIN	FORTAN	36 K	30	MREQ, HPU, LO, MALT, MATL	FOUT	Write match data on FAS
BINCOPY	COBOL	35 K	15	FOUT	FAS tape	Sort file and write FAS tape





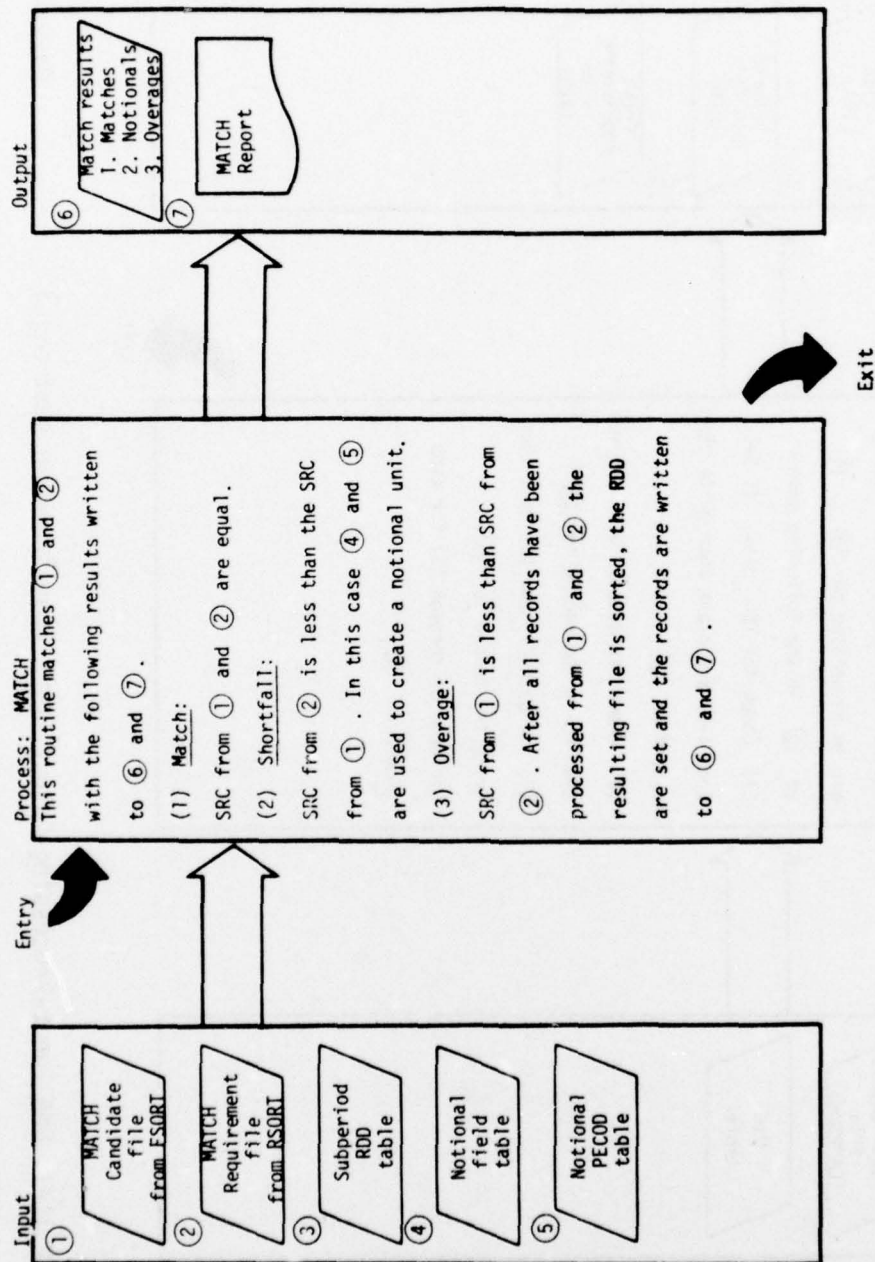


FIGURE II-1, FMA Input-Processing-Output Chart (continued on next page)

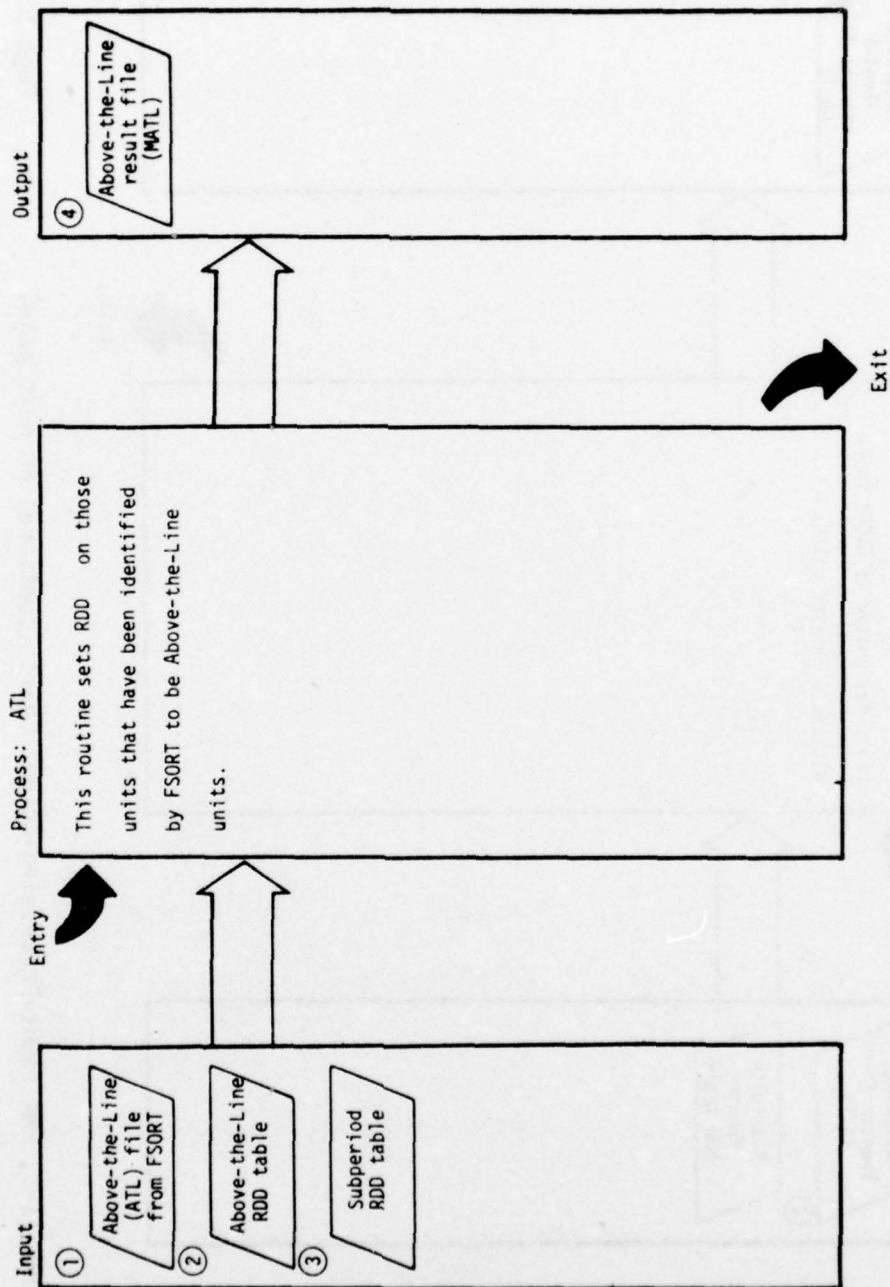


FIGURE II-1, FMA Input-Processing-Output Chart (continued on next page)

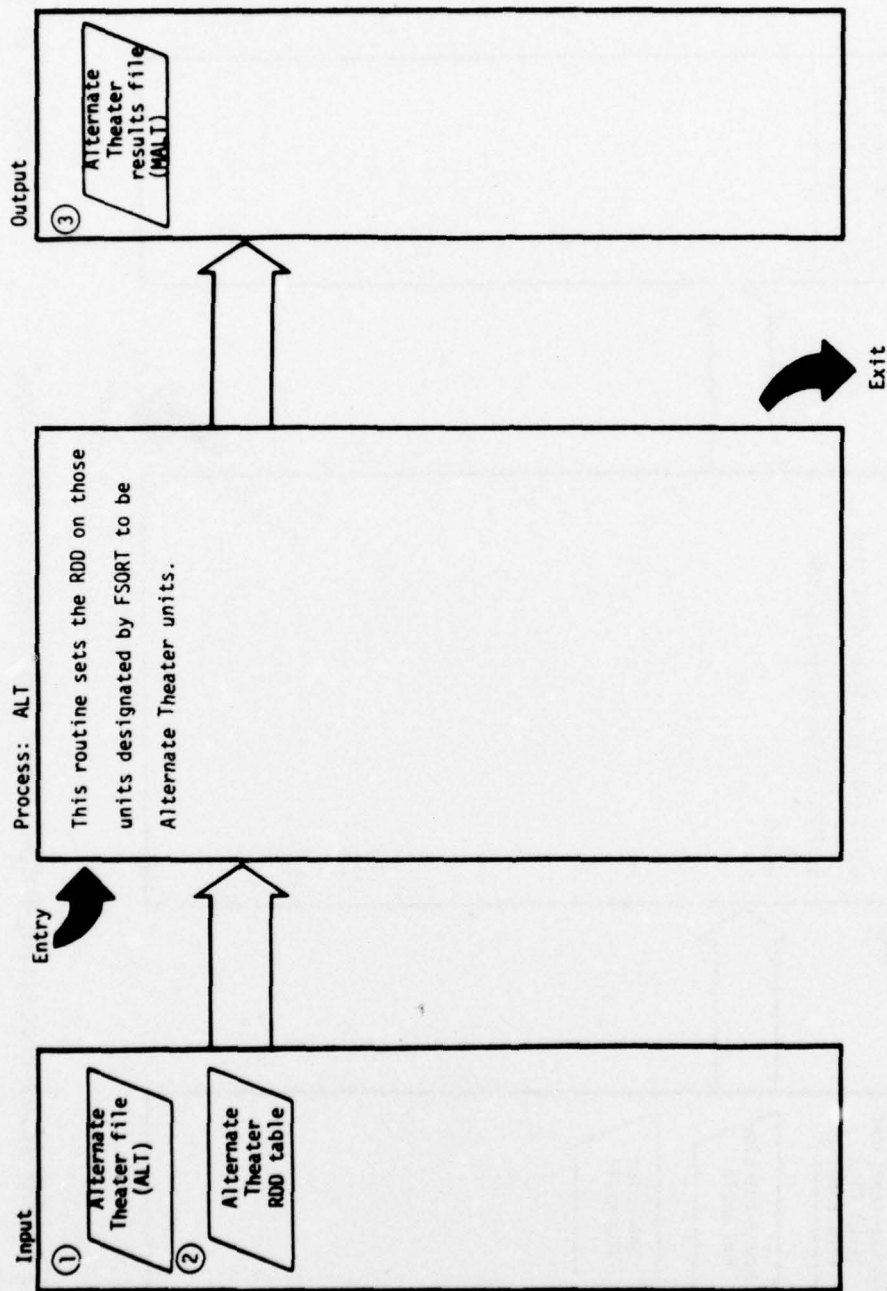


FIGURE II-1, FMA Input-Processing-Output Chart (continued on next page)

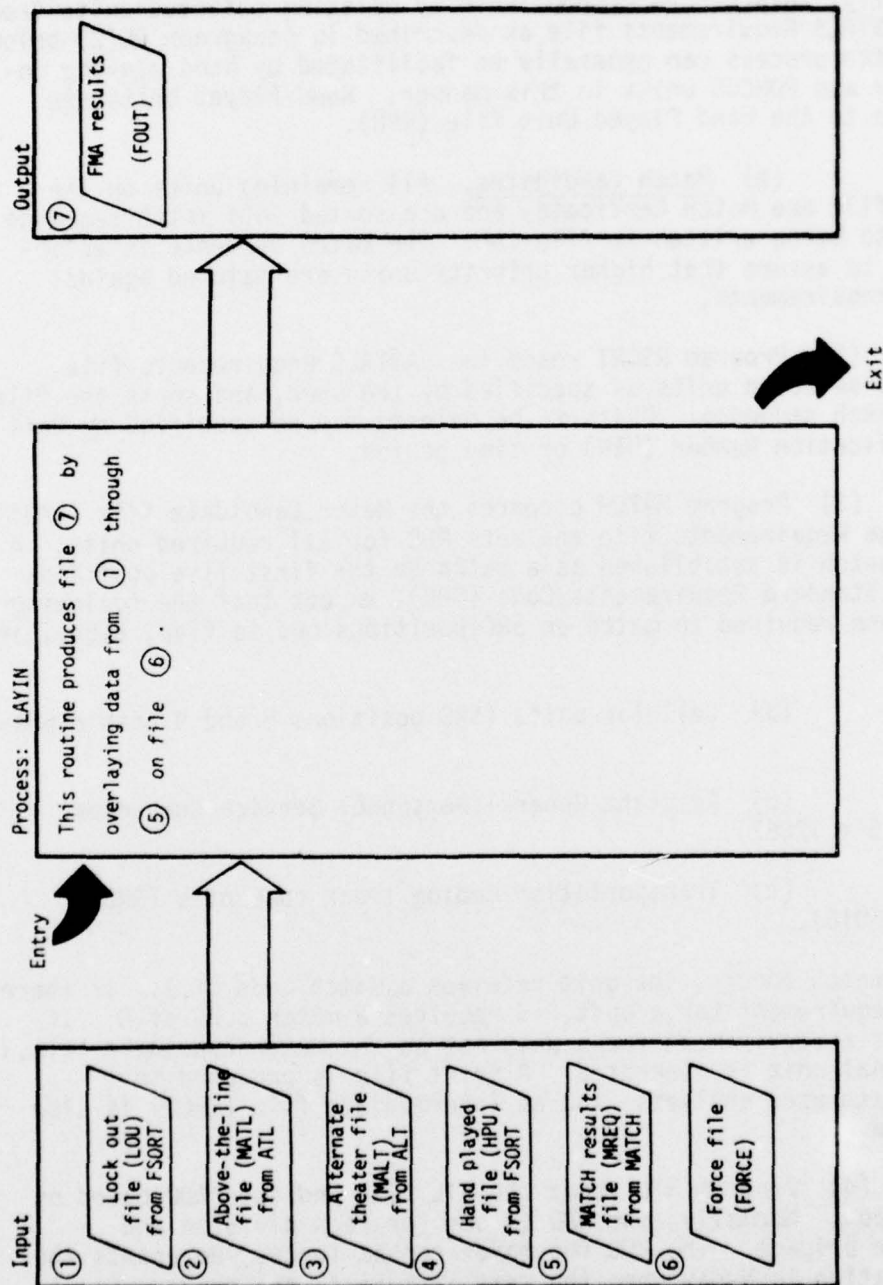


FIGURE II-1, FMA Input-Processing-Output Chart (concluded)

designated as required in the primary theater, so the user must assure that duplicate requirements are not generated by FASTALS. Generally, this can be accomplished by deleting selected units from the FASTALS Requirements file as described in paragraph 2b(2) below. The match process can generally be facilitated by hand playing in-country and POMCUS units in this manner. Hand Played Units are written to the Hand Played Unit file (HPU).

(e) Match Candidates. All remaining units on the Force file are Match Candidates and are sorted into match sequence prior to being written to file CAN. The match sequence is established to assure that higher priority units are matched against early requirements.

(2) Program RSORT reads the FASTALS Requirements file, deletes selected units as specified by the user, and sorts the file into match sequence. Units to be deleted may be specified by Unit Identification Number (UIN) or time period.

(3) Program MATCH compares the Match Candidate file (CAN) with the Requirements file and sets RDD for all required units. A valid match is established as a match on the first five positions of the Standard Requirements Code (SRC), except that the following units are required to match on SRC positions one to five, eight, and nine.

(a) Cellular units (SRC positions 8 and 9 both alphabetic).

(b) Adjutant General Personnel Service Companies (SRC 1-5 = 12067).

(c) Transportation medium truck companies (SRC 1-5 = 55018).

When a match occurs, the unit receives a match code of 1. If there is no requirement for a unit, it receives a match code of 0. If there is a requirement for a unit not on the Match Candidate file, a notional unit is generated. A print file is produced to facilitate user analysis, and an intermediate file (MREQ) is also produced.

(4) Program ATL reads the ATL file and sets RDD based on user input. Normally, one RDD is set for each division and separate brigade. The RDD for major combat forces represents the day relative to M-day when the unit arrives in the theater (port discharge date). This date is usually established based on input to a warfighting model. The time period is also set. Output file is file MATL.

(5) Program ALT sets RDD for units on file ALT based on user input. Output is file MALT.

(6) Program LAYIN merges the files LOU, MALT, MATL, HPU, and MREQ with the Force file to record the match results. A revised Force file is developed with match results overlaid on selected fields:

- (a) AUTH 1-3 = FASTALS UIN for matched or notional unit
= 000, overage
= blank, otherwise
- (b) AUTH 4-7 = FASTALS rules for matched or notional unit
= zero, otherwise
- (c) AUTH 8 = Match code, assigned as follows:
Required unit: 1
Not required: 0
Other, as specified for specific run
(FAS ADC01)
- (d) AUTH 9-10 = FASTALS time period for Required unit
= 99, overage
= blank, Lock Out or Alternate Theater
- (e) AUTH 11-13 = RDD for Required unit or Alternate Theater
= 999, overage
= blank, Lock Out
- (f) AUTH 14 = FASTALS Logical Region of Employment (LRE) for Required units
= zero for overage
= blank, otherwise
- (g) AUTH 15 = Theater code as assigned by Match.

c. Postprocessors

(1) Program BINCOPY reads and sorts the revised Force file (FOUT) to produce the final output tape in FAS format. If the primary tape is to be forwarded to the DA staff for analysis, the user should make a copy for retention at CAA.

(2) Program BUILD may again be used to interface CAMP with the Unit Data System (UDS) by creating a keyed access UDS file from the FAS tape. The UDS capabilities are described in Reference 1.

3. Interfaces with other Models. Much of CAMP utility results from its capability to interface with other models and systems. The FMA interfaces with three major models to provide match results. Requirements data are extracted from FASTALS; forces are extracted from FAS; output is fed back to FAS and into UDS for analysis.

a. FASTALS Interface. The FASTALS generates time phased support requirements for a given tactical scenario. A special program called CAMPSUM, written and maintained by the CAA Methodology and Resources Directorate, produces CAMP input from FASTALS output. Actually, there are three versions of CAMPSUM, each of which handles composite units (called '66' units because the SRC ends in '66') differently. Composite units are composed of a mix of teams and detachments and are tailored to meet specific requirements. There is no predetermined number or type of teams that make up a specific composite unit. However, FASTALS plays "type" composite units. The principal version of CAMPSUM is called 66 ROLLUP because the teams and detachments are rolled up to the level of the '66' SRC. In general, composite units on the FAS file do not have a '66' SRC but instead display the SRC of the header team. To facilitate matching, a new version of CAMPSUM (66 ROLLUP--MODIFIED) was developed to roll up composite units using total strengths but using the SRC of the header team. A third version (66 ROLLOUT) displays requirements for each team and detachment without using the composite unit designation. This version must be used if movement requirements are to be generated. CAMPSUM also accesses the table(s) of organization and equipment/troop program sequence number (TOE/PSN) file to append detail strength data and the TPSN to the FASTALS output.

b. FAS Interface. The FAS interface is the weakest link of CAMP. Concepts Analysis Agency does not control the FAS data base. The interface is in the form of manual transfer of tapes between ODCSOPS and CAA. Errors in FAS data create significant problems for CAMP. Three alternatives are open to the user to correct FAS data errors. A new tape can be requested from ODCSOPS, the existing tape can be loaded into UDS and updated by the user, or program SORTUDS can be revised to make the required data changes. The best alternative to choose depends on the time available and the nature of the changes.

c. UDS Interface. The UDS provides two major capabilities: the data base update capability discussed above and a generalized

report generator capability. The interface is provided by program BUILD. The UDS reports are of great value to the CAMP user in analyzing match results.

4. User Constraints

a. Data Base. A significant detractor of FMA utility is the lack of control over the FAS data base. Close coordination with ODCSOPS can eliminate many potential errors and correct errors that have already occurred.

b. SRC Substitutions. The SRC used in FASTALS do not correspond exactly with those used by FAS. Apparent mismatches result in generation of unnecessary notional units when in fact a similar unit is on the force but has a different SRC.

c. Composite Units. Actual composite units on the FAS file are tailored to meet specific requirements. The composition of such units may differ greatly from the type composition used by FASTALS. Thus, a given SRC may represent one type unit in FAS and another type unit in FASTALS. The best way to resolve this is by rolling out composite units to the team level. Efforts are underway to achieve this capability on the FAS file.

COMPUTER ASSISTED MATCH PROGRAM (CAMP)

CHAPTER III FUNCTIONAL DESCRIPTION - MOVEMENT REQUIREMENTS GENERATOR (MRG)

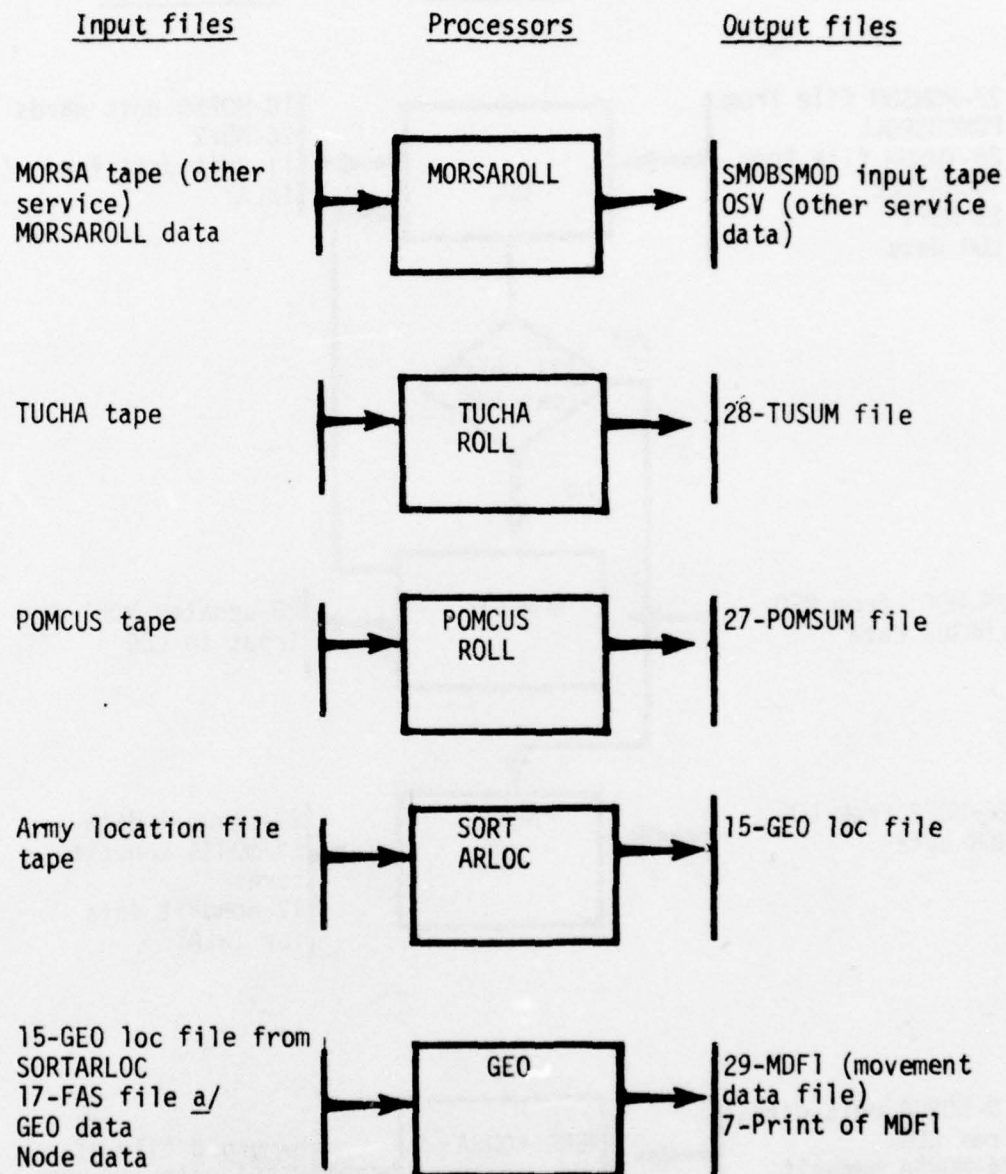
1. Introduction. The Movement Requirements Generator (MRG) interfaces FMA output with selected strategic mobility models. The MRG automates the production of input data for Strategic Mobility Simulation Model (SMOBSMOD) and the Mobility Requirements for Staff Analysis (MORSA) system. The MRG determines unit movement requirements for a force to be deployed. The force is entered in FAS format. The force may be entered directly from FMA; however, the more general occurrence is that FMA output is analyzed and reviewed by the DA staff prior to being entered in MRG. Actual unit locations are used as origins, and location codes required by the models are obtained automatically through an MRG interface with the Army Location file. Deployment weights for various categories of unit equipment are obtained automatically from the TUCHA file. A special file is used for those units that have equipment prepositioned in the theater. Accompanying supply weights are entered by the user. Nonunit movement requirements (fillers, replacement, and resupply) are calculated by MRG based on factors entered by the user. These factors include consumption rates, prepositioned quantities, and theater stockage objectives. After all movement requirements have been determined, the data are reformatted to meet input specifications for SMOBSMOD and MORSA. An automated interface allows the data (both unit and nonunit) to be entered into UDS, thus facilitating review and analysis.

2. System Processing Flow. The MRG consists of 13 programs as listed in Table III-1. Processing logic is shown in Figure III-1 and is discussed below.

a. Programs SORTARLOC, MORSAROLL, TUCAROLL, and POMCUSROLL are preliminary processes that build the data base actually used in subsequent MRG processes. Each of these programs reads a tape produced outside of CAA and creates a mass storage file with data required for subsequent processing. The formats for the tapes and output files are described in subsequent chapters. Use of these preliminary processes allows the user to analyze the input data thoroughly prior to making production runs. The computer resources required for production runs are thus considerably reduced because the output files are generally smaller than the input files and operate on time-saving mass storage files in lieu of tape files. In MORSAROLL, the user can specify how other service movement requirements are to be rolled up for SMOBSMOD. Origins and destinations may be rolled to specific SMOBSMOD nodes, and RDD and availability dates may be rolled into groups.

TABLE III-1, Movements Requirements Generator Programs

Program	Language	Size (words)	Time (minutes)	Function
SORTARLOC	COBOL	18 K	5	Produces Location file
MORSAROLL	COBOL	36 K	6	Rolls up other service movement data for SMOBSMOD
TUCHAROLL	COBOL	42 K	6	Produces TUCHA summary file
POMCUSROLL	COBOL	42 K	2	Produces POMCUS summary file
GEO	FORTRAN	58 K	16	Sets origin and destination codes
LOG	FORTRAN	34 K	28	Sets unit deployment weights. Produces MORSA cards
UPMDF1	FORTRAN	20 K	20	Allows SRC substitution
NUR	FORTRAN	38 K	11	Calculates nonunit movement requirements. Produces MORSA cards
PCKAGE	FORTRAN	36 K	14	Rolls movement requirements into SMOBSMOD packages
PRTPCK	FORTRAN	6 K	2	Prints report
MORSAMERGE	FORTRAN	6 K	12	Merges unit and nonunit MORSA cards
INLAY	FORTRAN	37 K	25	Lays movement data on FAS records
BUILDMRG	COBOL	16 K	2	Produces keyed access UDS file



a/Normally produced by FMA program LAYIN. May be produced by SORTUDS if no match processing is required.

FIGURE III-1, MRG Input-Processing-Output Chart (continued on next page)

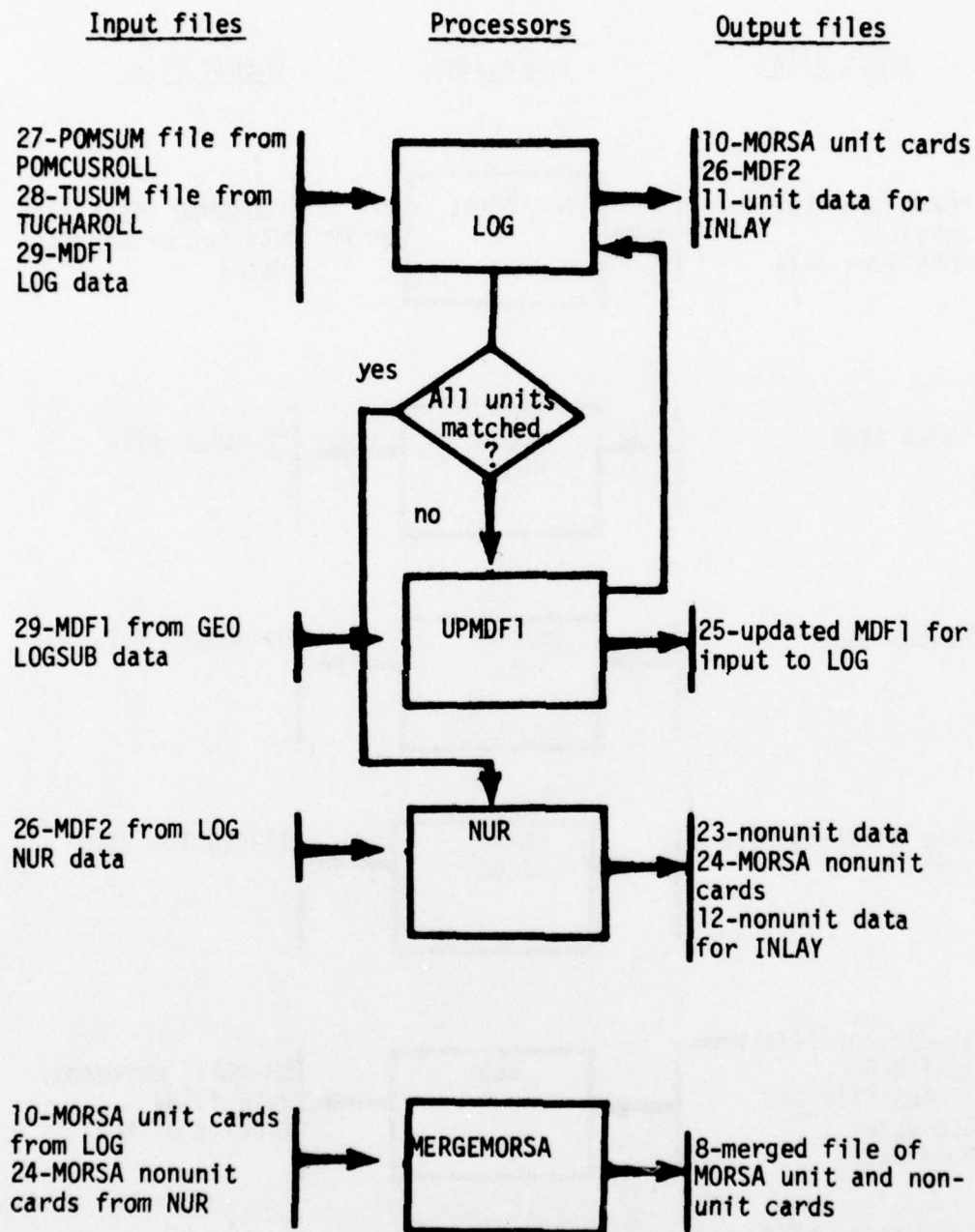
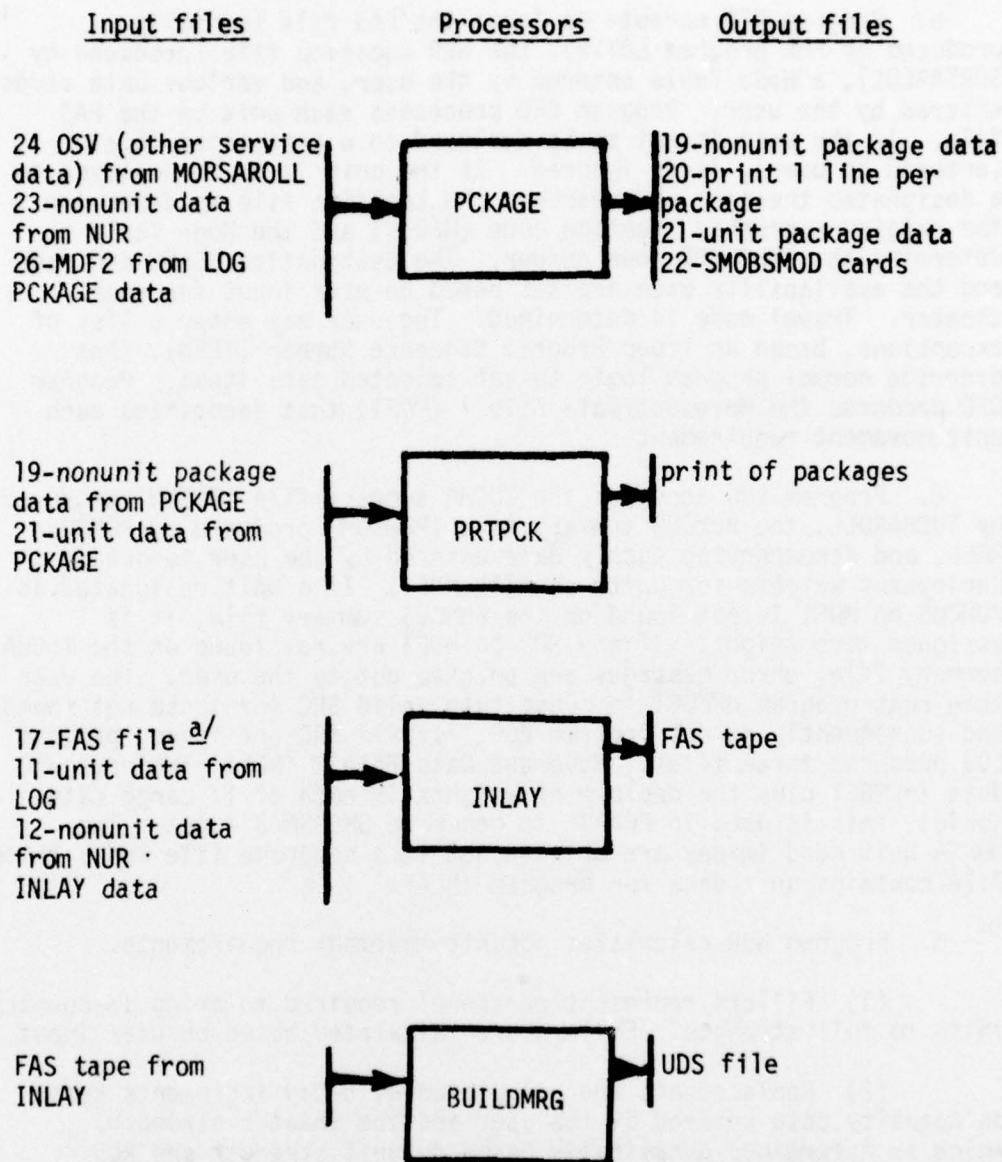


FIGURE III-1, MRG Input-Processing-Output Chart (continued on next page)



^{a/}Normally produced by FMA program LAYIN. May be produced by SORTUDS if no match processing is required.

FIGURE III-1, MRG Input-Processing-Output Chart (concluded)

b. Program GEO accepts as input the FAS file (normally produced by FMA program LAYIN), the GEO Location file (produced by SORTARLOC), a Node Table entered by the user, and various data cards entered by the user. Program GEO processes each unit on the FAS file. If the unit is not to be deployed to a designated theater (entered by user), it is ignored. If the unit is to be deployed to a designated theater, GEO searches the Location file to determine the origin geographic location code (GELOC) and the Node Table to determine the SMOBSMOD node number. The destination GELOC and node and the availability date are set based on user input for each theater. Travel mode is determined. The user may enter a list of exceptions, based on Troop Program Sequence Number (TPSN), that override normal program logic to set selected data items. Program GEO produces the Movement Data File 1 (MDF1) that identifies each unit movement requirement.

c. Program LOG accesses the TUCHA summary file (TUSUM) produced by TUCAROLL, the POMCUS summary file (POMSUM) produced by POMCUS-ROLL, and accompanying supply data entered by the user to obtain deployment weights for units on file MDF1. If a unit designated as POMCUS on MDF1 is not found on the POMCUS summary file, it is assigned zero weight. If any SRC on MDF1 are not found on the TUCHA summary file, error messages are printed out to the user. The user then runs program UPMDF1 to substitute valid SRC for those not found, and subsequently reruns program LOG. If all SRC are found, program LOG produces three files. Movement Data File 2 (MDF2) includes all data in MDF1 plus the deployment weights in each of 17 cargo categories; this is used in PCKAGE to generate SMOBSMOD input. The MORSA unit card images are written out to a separate file and a third file contains unit data for program INLAY.

d. Program NUR calculates nonunit movement requirements.

(1) Fillers represent personnel required to bring in-country units to full strength. Fillers are calculated based on user input.

(2) Replacements are calculated at 5-day increments based on casualty data entered by the user and the theater strength, which is determined dynamically based on unit strength and RDD information in MDF2 file.

(3) Resupply requirements are calculated for POL, ammunition, and dry bulk. For each theater, the user enters prepositioned quantities, consumption data, and theater stockage policies (day resupply starts, RDAY; stockage objective; and day objective is to be achieved, BLDUP). Program NUR simulates arrival of units and the arrival and consumption of supplies to determine resupply requirements. For all days prior to RDAY, the model generates

resupply requirements to maintain a safety level (entered by the user). For each 5-day interval between RDAY and BLDUP, the model interpolates between the on-hand level at RDAY and the stockage objective at BLDUP to determine requirements. However, if RDAY level is greater (in terms of days of supply) than the stockage objective, resupply requirements are not generated until the level falls below the objective. For each 5-day increment after BLDUP, the model generates resupply requirements to maintain the stockage objective. Nonunit requirements are written to a MORSA card file and a file containing data for program INLAY.

e. Program MERGEMORSA produces a single set of MORSA card images on tape by merging the MORSA files produced by LOG and NUR.

f. Programs PCKAGE and PRTPCK produce SMOBSMOD input cards and print out the results for user review. Both a detailed and a summary report are produced. Movement requirements are rolled into packages. All requirements with the same RDD, availability date, origin node, destination node, and travel mode are rolled into one package. The user may obtain a higher degree of rollup by specifying input parameters that cause several RDD and several availability days to be packaged together such that the package will have the earliest RDD and latest availability day of the group.

g. Program INLAY creates a FAS file tape overlaid with movement data. New FAS records are created for nonunit requirements; the user enters the FICOD, COMPO and EDATE, and the UIC (first position N) is set sequentially by INLAY. Units on the FAS file that are not deployed are not overlaid with movement data.

h. Program BUILDMPG reads the FAS tape created by INLAY and produces a keyed access UDS file.

3. Interfaces with Other Models. The MRG is designed specifically to interface the FAS with SMOBSMOD and MORSA. Interfaces with additional mobility models could be programed if required by the user and if all required data items are available in CAMP files. Requirements for additional interfaces should be coordinated with the CAA Methodology and Resources Directorate. The MRG also interfaces with FAS and UDS.

a. The MRG/SMOBSMOD interface provided by program PCKAGE results in production of SMOBSMOD input card images in a mass storage file. The SMOBSMOD input requirements are described in Chapter VI and overall SMOBSMOD capabilities are described in Reference 2.

b. The MRG/MORSA interface results in production of MORSA

input card images on tape. Concepts Analysis Agency does not currently have the capability to process MORSA data, but is required by various study directives to produce the data for processing at the Joint Chiefs of Staff (JCS) level. Input to MORSA is described in Chapter VI and is produced by programs LOG and NUR and merged by program MERGEMORSA. The MORSA system is operated and maintained at the Command and Control Technical Center (CCTC), Logistics Data System Division, and is somewhat volatile. The user is cautioned to coordinate with CCTC prior to running MRG to ascertain if any MORSA input requirements have changed. The MORSA system is described in Reference 3.

c. The MRG/UDS interface provided by program INLAY results in a significant report generation capability. The UDS file as created by MRG is described in Chapter VI. Overall UDS capabilities are described in Reference 1.

d. An MRG/FAS interface can be provided by program SORTUDS to allow use of an actual force (with or without notional units) as a basis for determination of movement requirements without prior FMA processing. In this case, the force must contain unit RDD in the ADC03 field.

4. User Constraints

a. Data Base. Inconsistencies occur relatively frequently in the various files accessed by MRG. The user should review the data contained in all computer tapes obtained outside of CAA. Errors and inconsistencies should be coordinated with the source of the data.

b. MORSA Output. Concepts Analysis Agency does not have the capability to process MORSA data. The MORSA system is operated by CCTC on the JCS Worldwide Military Command and Control System (WWMCCS) computer, and CAA is required to provide movement data in MORSA format. However, correction of errors is difficult and often requires that CAA rerun the job.

COMPUTER ASSISTED MATCH PROGRAM (CAMP)

CHAPTER IV

DATA DICTIONARY

<u>Data element</u>	<u>Description</u>	<u>Set</u>	<u>Used</u>	<u>Remarks</u>
A0	Weight of non-air transportable (NAT) vehicles.	Input from TUCHA file for each SRC. Input from POMCUS file for each POMCUS UIC (tenths of short tons).	TUCHAROLL POMCUSROLL LOG	Rolled up for output to MORSA. Output to SMOBSMOD cargo category 2.
A1	Weight of outside vehicles.	Input from TUCHA file for each SRC. Input from POMCUS file for each POMCUS UIC (tenths of short tons).	TUCHAROLL POMCUSROLL LOG	Rolled up for output to MORSA. Output to SMOBSMOD cargo category 5.
A2C	Weight of oversize containerizable vehicles.	Input from TUCHA file for each SRC. Input from POMCUS file for each POMCUS UIC (tenths of short tons).	TUCHAROLL POMCUSROLL LOG	Rolled up for output to MORSA. Output to SMOBSMOD cargo category 10.
A2D	Weight of oversize noncontainerizable vehicles.	Input from TUCHA file for each SRC. Input from POMCUS file for each POMCUS UIC (tenths of short tons).	TUCHAROLL POMCUSROLL LOG	Rolled up for output to MORSA. Output to SMOBSMOD cargo category 11.
A3C	Weight of containerizable bulk vehicles.	Input from TUCHA file for each SRC. Input from POMCUS file for each POMCUS UIC (tenths of short tons).	TUCHAROLL POMCUSROLL LOG	Rolled up for output to MORSA. Output to SMOBSMOD cargo category 14.

<u>Data element</u>	<u>Description</u>	<u>Set</u>	<u>Used</u>	<u>Remarks</u>
ACAMO	Weight of accompanying ammunition.	LOG	NUR	Output to MORSA (short tons) and SMOBSMOD (category 17)(thousand short tons).
ACTCO	Action code. See data element EDATE.	Input from FAS file.		Identifies the type change being made on a given EDATE. E.g., ACTCO = J implies deactivation of a unit.
ADCON				
ADC01	See data element Theater Code.			
ADC02	See data element Time Period.			
ADC03	See data element RDD.			
AMMO	Ammunition weight.	NUR for resupply.		Output to MORSA and SMOBSMOD (category 17).
AREA	Area abbreviation. A three-position code which identifies a specific geographic area. See data elements LOCCO and MBLOC.	Input from Army location file.	GEO	DA Cir 525-10-4, -3. CONUS areas are identified by Army area and state. Example: 1VA implies First Army, Virginia. Overseas areas are generally identified by an abbreviation of the country name. Example: FRA for France.

<u>Data element</u>	<u>Description</u>	<u>Set</u>	<u>Used</u>	<u>Remarks</u>
ARLOC	Army location code. A unique five position code which identifies a geographic location at which Army activities may be situated or which may have potential military significance.	Input from Army location file.	SORTARLOC	
ASGMT	Assignment. Identifies major command to which a unit is assigned.	Input from FAS file.		Manually set for notional units.
ASPLY	Weight of accompanying supply (excluding ammunition).	NUR		Output to MORSA and SMOBSMOD.
AUTHR				
AUTHR 1-3	See data element UIN.			
AUTHR 4-7	See data element Rules.			
AUTHR 8	See data element Match Code.			
AUTHR 9-10	See data element Time Period.			
AUTHR 11-13	See data element RDD.			

<u>Data element</u>	<u>Description</u>	<u>Set</u>	<u>Used</u>	<u>Remarks</u>
AUTHR 14	See data element LRE.			
AUTHR 15	See data element Theater Code.			
AVAIL	Availability date. Day relative to the starting day of a scenario on which a given unit is available for deployment.	GEO		Output to MORSA and SMOBSMOD.
B0	Square feet of NAT non-self deployable aircraft (NSDA).	Input from TUCHA file for each SRC. Input from POMCUS file for each POMCUS UIC.		Rolled up for output to MORSA. Output to SMOBSMOD category 3.
B1	Square feet of outsize NSDA.	Input from TUCHA file for each SRC. Input from POMCUS file for each POMCUS UIC.		Rolled up for output to MORSA. Output to SMOBSMOD category 4.
B2C	Square feet of containerizable oversize NSDA.	Input from TUCHA file for each SRC. Input from POMCUS file for each POMCUS UIC.		Rolled up for output to MORSA. Output to SMOBSMOD category 8.
Branch	Branch of service for TOE units.	Input from FAS file and FASTALS.	N/A	Example: EN for engineers.

<u>Data element</u>	<u>Description</u>	<u>Set</u>	<u>Used</u>	<u>Remarks</u>
BULK	Cargo having dimensions not larger than 104 x 84 x 96 inches.	LOG		Output to MORSA.
CARRS	Combat arms regimental system. Historical designation assigned to combat arms units.	Input from FAS file.	N/A	
COMPO	Component code. Identifies unit status.	Input from FAS file.	FSORT	1 = Active 2 = National Guard 3 = Reserve 4 = Notional.
DAMPL	DA master priority list. Priority grouping of units for allocation of personnel and/or equipment.	Input from FAS file.	FSORT	Match sequence is keyed on DAMPL.
DSCMP	Display/compute indicator.	Input from FAS file.	SORTUDS	D0 implies display only (a titular record). DC implies display and compute (i.e. count this unit's strength).
Desti- nation	The location to which a unit is being deployed.	GEO. Based on Theater Code.	NUR	Output to MORSA and SMOBSMOD.

<u>Data element</u>	<u>Description</u>	<u>Set</u>	<u>Used</u>	<u>Remarks</u>
Dry Re-supply	Quantity of resupply in all classes except III (POL) and V(ammo) (thousands of tons).	NUR		SMOBSMOD cargo category 16.
EDATE	Effective date of a transaction. See data element ACTCO.	Input from FAS file.		Manually set for notional units.
FCTCD	Foreign Country Code Description. The full name of foreign countries.	Input from Army location file.	N/A.	DA Cir 525-10-4.
FICOD	Force identification code. Differentiates between various planning forces.	Input from FAS file.		Example: J = JSOP force T = Total Force.
FPLAN	Force Planning Code. Classification code for strategic and force planning.	Input from FAS file.	N/A	First position: A = Division forces B = Special mission C = General support.
GELOC	Specified geographic location code. A JCS established equivalent to the ARLOC covering locations for all military activities.	Input from Army location file.	GEO	Output to MORSA.

<u>Data element</u>	<u>Description</u>	<u>Set</u>	<u>Used</u>	<u>Remarks</u>
J1	Quantity of non-vehicular outside cargo.	Input from TUCHA file for each SRC. Input from POMCUS file for each POMCUS UIC (tenths of short tons).	LOG	Rolled up for output to MORSA (short tons). Output to SMOBSMOD category 7 (thousand short tons).
J2C	Quantity of non-vehicular oversize containerizable cargo.	Input from TUCHA file for each SRC. Input from POMCUS file for each POMCUS UIC (tenths of short tons).	LOG	Rolled up for output to MORSA (short tons). Output to SMOBSMOD category 13 (thousand short tons).
J2D	Quantity of non-vehicular oversize noncontainerizable cargo.	Input from TUCHA file for each SRC. Input from POMCUS file for each POMCUS UIC (tenths of short tons).	LOG	Rolled up for output to MORSA (short tons). Output to SMOBSMOD category 12 (thousand short tons).
J3C	Quantity of non-vehicular bulk containerizable cargo.	Input from TUCHA file for each SRC. Input from POMCUS file for each POMCUS UIC (tenths of short tons).	LOG	Rolled up for output to MORSA (short tons). Output to SMOBSMOD category 15 (thousand short tons).
J7C	Quantity of accompanying dry bulk supply.	Input from TUCHA file.	N/A	
JCSTY	JCS type unit code. JCS designated equivalent of SRC.	Input from FAS and TUCHA files.	N/A	Output to MORSA.

<u>Data element</u>	<u>Description</u>	<u>Set</u>	<u>Used</u>	<u>Remarks</u>
Level	First three characters of the FAS unit description. Denotes the organizational level of the unit.	Input from FAS and FASTALS files.	N/A	Example: TM = team BN = battalion Note: Not to be confused with level at position 10 of SRC.
LFLLC	Geographic coordinates. Latitude and longitude.	Input from Army location file.	N/A	
LOCCO	See data element AREA. The area code for the unit home station.	Input from FAS file.	GEO	
LOCNA	Location name abbreviated. Nine characters maximum.	Input from Army location file.	GEO	
LOCNM	Location name; 17 characters maximum.	Input from Army location file.	GEO	
LRE	Logical Region of Employment.	Input from FASTALS.		1 = combat zone 2 = corps forward 3 = corps rear 4 = commz forward 5 = commz rear 6 = offshore Output to FAS.
Match Code	Indicator for match results.	MATCH	LAYIN	1 = required 0 = excess X = locked out.

<u>Data element</u>	<u>Description</u>	<u>Set</u>	<u>Used</u>	<u>Remarks</u>
MBLOC	See data element AREA. Area code for unit mobilization station.	Input from FAS file.	GEO	
MBSTA	See data element LOCNA. Abbreviated location name of unit mobilization station.	Input from FAS file.	GEO	
NAT	Non-air transportable. Cargo having dimensions larger than 1453 x 144 x 156 inches or 1453 x 216 x 114 inches.	LOG		Output to MORSA.
NSDA	Non-self-deployable aircraft.	LOG		Output to MORSA.
Origin	Location at which a unit or package originates deployment.	GEO. Usually home station for active units, mobilization station for nonactive units.		Output to MORSA as the GELOC. Output to SMOBSMOD as a node number.
OUTSZ	Outsize. Cargo having dimensions larger than 810 x 117 x 105 inches but not excluding the limits of air transportability.	LOG		Requires C5A aircraft. Output to MORSA.

<u>Data element</u>	<u>Description</u>	<u>Set</u>	<u>Used</u>	<u>Remarks</u>
OVRSZ	Overize. Cargo having dimensions larger than 104 x 84 x 96 inches but no larger than 810 x 117 x 105 inches.	LOG		Requires C-141 aircraft. Output to MORSA.
PAX	Passengers.	Input from TUCHA.	NUR LOG	Output to MORSA and SMOBSMOD.
PECOD	Program element code. Budget code which groups similar forces.	Input from FAS file.		Manually set for notional units.
PHASE	Authority for a FAS record entry.	Input from FAS file.		Manually set for notional units.
RD	Record description. Code to identify the type cargo. See data element RT.	NUR LOG		Output to MORSA. A = ammo F = fillers G = resupply dry or replacements N = NSDA 8 = POL.
RDD	Required delivery date. Day relative to initial day of a scenario when unit or package is required to arrive in the destination theater (port discharge date).	MATCH ATL ALT	NUR	Output to MORSA and SMOBSMOD.

<u>Data element</u>	<u>Description</u>	<u>Set</u>	<u>Used</u>	<u>Remarks</u>
ROBCO	Ostensibly, a code to identify reserve component units or part of a force grouping. Now used extensively to designate POMCUS packages.	Input from FAS file.	FSORT LOG	Output to MORSA.
RT	Record type code to identify the type cargo. See data element RD.	NUR LOG		Output to MORSA U = unit P = POL S = other.
Rules	Zero/one variables to indicate reason for a unit requirement. Manual play, theater structure, existence and/or workload.	Input from FASTALS.		Output to FAS.
SRC	Standard Requirements Code. Identifies type unit (TOE).	Input from FAS, FASTALS and TUCHA.	FSORT RSORT MATCH LOG	Pos 1-2 branch identity 2-5 type organization 6 series 7 year 8-9 variation 10 level 11-12 paragraph. Manually set for notional units.
STACO	Station code. Army location code for unit home station. See ARLOC.	Input from FAS file.		

<u>Data element</u>	<u>Description</u>	<u>Set</u>	<u>Used</u>	<u>Remarks</u>
STAGR	Structured aggregate strength. Full TOE strength.	Input from FAS. Input from FASTALS for use in generation of notional units.	LOG NUR MATCH	Output to FAS.
STENL	Structured enlisted strength.	Input from FAS. Input from FASTALS for use in generation of notional units.	MATCH	Output to FAS.
STNNM	See data element LOCNA. Location name (abbreviated) of unit's home station.	Input from FAS.	GEO	
STOFF	Structured officer strength.	Input from FAS. Input from FASTALS for generation of notional units.	MATCH	Output to FAS.
STWOF	Structured warrant officer strength.	Input from FAS. Input from FASTALS for generation of notional units.	MATCH	Output to FAS.
Theater Code	One character code designating the theater to which a unit is deploying.	LAYIN	GEO NUR	

<u>Data element</u>	<u>Description</u>	<u>Set</u>	<u>Used</u>	<u>Remarks</u>
Time Period	Time increment established in FASTALS.	Input from FASTALS.	RSORT MATCH	Output to FAS. RDD are based on Time Period as determined for a specific scenario.
TLAC	Type location or activity code.	Input from ARLOC.	SORTARLOC	Example: AIN = Army installation. See App B, DA Cir 525-10-4.
TMODE	Preferred mode of travel.	LOG NUR		Output to MORSA and SMOBSMOD. MORSA SMOBSMOD Air A 1 Sea S 2 Optional P 0 Mixed - 3
TPSN	Troop program sequence number. Code which groups units by type and size.	Input from FAS. Input from FASTALS for generation of notional units.		Output to FAS. See AR 18-19 for explanation.
TYPCO	Type code.	Input from FAS file.	FSORT	1 = TOE unit 2 = TDA augmentation 3 = TDA unit.
UGRID	Universal transmercator grid coordinates (military grid coordinates).	Input from ARLOC.	N/A	

<u>Data element</u>	<u>Description</u>	<u>Set</u>	<u>Used</u>	<u>Remarks</u>
UIC	Unit identification code. Code which uniquely identifies each unit.	Input from FAS. Input from POMCUS, generated by MATCH for notional units.	LOG	Output to FAS.
UIN	Unit identity number. Associates SRC with LRE within FASTALS.	Input from FASTALS	RSORT	Output to FAS.
ULCCC	See Level.			
UNMBR	Unit number. The numerical portion of the unit designation.	Input from FAS generated by MATCH for notional units.		Output to FAS.
UNTDS	Unit description.	Input from FAS and FASTALS.		Output to MORSA.

COMPUTER ASSISTED MATCH PROGRAM (CAMP)

CHAPTER V JOB PREPARATION

1. FMA Job Preparation

a. General. To run an FMA job, the user must obtain two basic input files, FAS and CAMPSUM, and prepare eleven manual input tables. The files and tables are described below.

b. Input Files

(1) FAS file. The FAS file is fully described in Reference 4. The file contains one or more records for each unit in the force. Multiple records occur when a unit changes status (e.g. receives an updated SRC), in which case the effective date of the change is shown on the applicable record. Each record is 307 characters. Fields applicable to CAMP have been described in Chapter IV and are shown in Table V-1. The ADCON and AUTHR fields are generally overlaid with other data. The ADCON field is used by the DA staff to pass selected data to CAA for match purposes. The ADC01 field contains the Theater Code which must be used in match processing as it identifies the theater to which a unit is eligible to deploy. The ADC02 and ADC03 fields may be used by the DA staff to identify a specific Time Period and RDD: in general, however, these values will not be specified by DA but are set during match processing. Match results are forwarded to DA in the AUTHR field.

(2) CAMPSUM. The FASTALS Requirements file is provided in CAMPSUM format as described in Table V-2. The file contains one record for each type unit requirement, except that each above-the-line unit is represented by its Headquarters Company. The FASTALS system is described in Reference 5.

c. Preliminary Processing. The FAS file is loaded into UDS using program BUILD. The user should analyze the force and make any required updates with UDS; then a UDS extract is performed to obtain the desired force for matching purposes. Program SORTUDS is then used to create the mass storage FAS file used by FMA. Once this file is obtained, any number of matches can be made against it.

d. Input Tables. FMA programs require the input tables described below. Normally, a different set of tables is developed by the user for each match.

TABLE V-1, FAS File Record

<u>Position</u>	<u>Data element</u>
2	FICOD
3	COMPO
4-9	UIC
10-15	EDATE
16	ACTCO
27-28	ASGMT
29-43	AUTHR
44-45	Branch
46-47	CARSS
58-62	DAMPL
71-72	DSCMP
80-82	FPLAN
83-87	JCSTY
88-90	LOCCO
96-98	MBLOC
101-109	MBSTA
119-124	ADCON
127	PHASE
130-132	ROBCO
134-145	SRC
147-151	STACO
154-162	STNNM
180-184	TPSN
186	TYPKO
194-197	UNMBR
198-203	PECOD
204-224	UNTDS
278-282	STOFF
283-287	STWOF
288-292	STENL
293-297	STAGR

TABLE V-2, CAMPSUM File Record

<u>Position</u>	<u>Type</u>	<u>Description</u>
1-3	N	FASTALS UIN
6-14	AN	SRC position 1-9
15-16	N	SRC paragraph
17	N	Personnel level
18	N	Equipment level
20-21	A	Branch
23-54	A	Unit description
56-58	N	FASTALS strength
60-61	N	FASTALS Time Period
64-67	N	FASTALS Rules
		Manual play
		Theater structure variable
		Existence rule
		Workload
69	N	FASTALS Logical Region of Employment
71	N	Combination rule <u>a/</u>
73	N	Rounding rule <u>a/</u>
75-79	N	TPSN
81-85	N	Officer strength <u>b/</u>
87-91	N	Warrant Officer strength <u>b/</u>
99-103	N	Enlisted strength <u>b/</u>
105-109	N	Total strength <u>b/</u>

a/ Not used in CAMPb/ Used as structured strength for notional units

(1) RSORT input. Two card types are input to RSORT to cause various records to be deleted from the CAMPSUM file. Normally these records will correspond to units which are played by hand in FSORT. Card type 1 is described in Table V-3. Exactly one of this card type is required. All records with Time Period less than the input value (First Time Period) will be deleted from the CAMPSUM file. For example, if First Time Period equals two, then all units with Time Period = 1 (usually corresponding to in-country units) will be deleted. Card type 2 is described in Table V-4. One card type 2 is entered for each UIN to be deleted. All records with a specified UIN and Time Period later than or equal to the specified UIN will be deleted.

(2) FSORT input. Six input tables are required for FSORT.

(a) The Hand Played UIC data as described in Table V-5 are used to change selected data fields on a FAS record. One card is entered for each unit to be changed. If RDD is not specified, the specified fields will be changed, and the unit will be processed according to the changed data; the unit will not be processed to the Hand Played Unit file (HPU). If a new RDD is specified, the unit will be hand played by FMA.

(b) The Lock Out ADC01 data described in Table V-6 are used to specify theaters not to be considered in this match. One card is entered for each theater to be locked out. Units with specified Theater Code are not match candidates in FMA and are not deployed in MRG.

(c) The Alternate Theater ADC01 data described in Table V-7 are used to designate alternate theaters. One card is entered for each Theater Code representing an alternate theater. Units with a specified Theater Code are not considered for matching in FMA but are deployed in MRG. This table is also input to Program ALT.

(d) MCODE 1 (POMCUS) ROBCO data. One card is entered for each POMCUS ROBCO as described in Table V-8. POMCUS units are hand played.

(e) The MCODE 0 (in-country) data require two card types as described in Table V-9. In-country units are hand played.

(f) The Lock Out COMPO data described in Table V-10 are used to delete entire sets of units from the force.

TABLE V-3, RSORT Card Type 1

<u>Position</u>	<u>Format</u>	<u>Data entry</u>
1-3	I3	First Time Period

TABLE V-4, RSORT Card Type 2
(Lock Out UIN)

<u>Position</u>	<u>Format</u>	<u>Data entry</u>
1-3	I3	FASTALS UIN
5-6	I2	Time Period

TABLE V-5, Hand Played UIC Data

<u>Position</u>	<u>Type</u>	<u>Data entry</u>
1-6	A	UIC--Unit Identification Code of Force record to be changed or hand played.
8-10	A	New ROBCO (blank if no change). Appears in final output.
12	A	New Theater Code (blank if no change). Appears in AUTHR field of final product.
14-15	N	Hand Played Time Period (if blank, program will calculate correct time period to correspond to RDD).
17-19	N	Hand Played RDD. If this field is entered the unit will be hand played; if this field is blank, the unit will be processed normally in accordance with other changes.

TABLE V-6, Lock Out ADC01 Data

<u>Position</u>	<u>Type</u>	<u>Data entry</u>
1	A	ADC01--Theater Code. Locked out in FMA and not deployed in MRG.

TABLE V-7, Alternate Theater ADC01 Data

<u>Position</u>	<u>Type</u>	<u>Data entry</u>
1	A	ADC01--code to select alternate theater support units (TPSN less than 20000)
3-5 <u>a/</u>	N	Earliest RDD to assign to units with this ADC01
7-9 <u>a/</u>	N	Last RDD to assign to units with this ADC01
11-13 <u>a/</u>	N	Increment between previous values to assign RDD
15-17 <u>a/</u>	A	AUTHR 15--Theater Code to be used by MRG

a/these fields are used by ALT but not by FSORT

TABLE V-8, MCODE 1 (POMCUS) ROBCO Data

<u>Position</u>	<u>Type</u>	<u>Data entry</u>
1-3	A	ROBCO field values to be considered POMCUS unit--the asterisk character (*) may be used to indicate a don't care condition for any position of the field. Only COMPO 1 units are considered.
5-7	N	RDD to be assigned above-the-line units with this ROBCO. May be blank in which case RDD will be assigned by TPSN as in Table V-14.
9-11	N	RDD to be assigned to support units with this ROBCO.

TABLE V-9, MCODE 0 (In-country) Data

<u>Position</u>	<u>Type</u>	<u>Data entry</u>
1-3 <u>a/</u>	N	M-day for primary theater. RDD for in-country units.
1	A	ADC01 value to be considered in-country in the primary theater. One card for each ADC01 value.

a/first card only

TABLE V-10, Lock Out Compo Data

<u>Position</u>	<u>Type</u>	<u>Data entry</u>
1	N	COMPO value to be locked out. Records with these COMPO values are deleted from the force file.

(3) MATCH input. Three input tables are required by MATCH.

(a) The Subperiod RDD data described in Table V-11 are used to designate RDD to be assigned within each Time Period. Higher priority units within a given Time Period are assigned the earlier RDD within that Time Period. This table is also used by ATL to assign a Time Period to each above-the-line unit.

(b) Tables V-12 and V-13 are used to create FAS records for notional units. Normally the values to be entered are specified by the DA staff.

(4) ATL input. ATL requires the input described in Table V-14. One card is entered for each TPSN. RDD are normally determined by the DA staff.

2. MRG Job Preparation

a. General. To run an MRG job, the user must obtain the five basic input files (ARLOC, TUCHA, POMCUS, MORSA and FAS) and provide card image input to six programs. The ARLOC file is obtained quarterly from US Army Management Systems Support Agency. The file is documented in References 6 and 7. The TUCHA file used is the Army input to the JCS TUCHA file used in the Joint Operations Planning System (JOPS). The JCS TUCHA file is documented in Reference 8. This file is obtained from the United States Army Command and Control Support Agency (USACCSA) quarterly. The POMCUS file is a nonstandard file provided to CAA on request by ODCSLOG. The MORSA file is obtained on request from CCTC through JCS-J4. See Reference 3 for documentation. Note that this file contains the MORSA data record, not the MORSA transaction card image which is produced by CAMP. The FAS file is produced by FMA Program LAYIN. The files and input data are described in subsequent paragraphs.

b. Preprocessors. MRG requires four preprocessor programs identified in Chapter III to process the input tapes. These programs are required only when the tapes to be processed are updated. The user is encouraged to coordinate with Joint and Strategic Forces Directorate to determine the status of the preprocessor output files. If the files are current, the preprocessor runs can be eliminated.

(1) ARLOCSORT. This preprocessor produces the GEO file from the ARLOC file. The layout of the ARLOC file is shown in Table V-15. Manual inputs are not required.

TABLE V-11, Subperiod RDD Data

<u>Position</u>	<u>Data entry</u>
1-4	RDD 1st subperiod
5-8	RDD 2nd subperiod
9-12	RDD 3rd subperiod
13-16	RDD 4th subperiod
17-20	RDD 5th subperiod
21-24	RDD 6th subperiod
25-28	last day of time period

One card per time period-10 cards required

TABLE V-12, Notional Unit Fields

<u>Card</u>	<u>Position</u>	<u>Type</u>	<u>Data entry</u>
1	1	A	FICOD
2	1-6	N	EDATE
3	1	A	ACTCO
4	1-2	A	ASGMT
5	1	N	AUTH 8
6	1-2	N	CARSS
7	1-2	A	DSCMP
8	1-3	A	FPLAN
9	1-3	AN	LOCCO
10	1-3	AN	MBLOC
11	1-6	A	MBSTA pos 1-6
12	1-3	A	MBSTA pos 7-9
13	1	A	PHASE
14	1-5	AN	STACO
15	1-6	A	STNNM pos 1-6
16	1-3	A	STNNM pos 7-9
17	1	N	TYPCO
18	1	N	COMPO
19	1-2	AN	UIC pos 1-2
20	1-2	AN	UIC pos 1-2 host nation support unit (not currently used)

TABLE V-13, Notional Unit PECOD Data

<u>Position</u>	<u>Type</u>	<u>Data entry</u>
1-2	N	SRC pos 1-2
5-10	AN	PECOD value to be assigned

TABLE V-14, Above-the-Line RDD Data

<u>Position</u>	<u>Type</u>	<u>Data entry</u>
1-5	N	TPSN
8	N	Theater Code to be assigned
11-13	N	RDD to be assigned

TABLE V-15, ARLOC File Record Description

<u>Data name</u>	<u>Description</u>	<u>Start pos</u>	<u>No char</u>	<u>Type</u>
AREA	Geographic area	1	3	A/N
ARLOC	Army location code	9	5	A/N
GELOC	JCS geographic location code	18	4	A
LOCNA	Abbreviated location name	27	9	A/N
LOCNM	Location name	40	17	A/N
TLAC	Type installation	65	3	A
FCTCD	State or foreign country	78	18	A/N
UGRID	Universal transverse mercator grid coordinates	98	11	A/N
LFLLC	Latitude/longitude	112	15	A/N
STAT	Status of record	129	1	A

(2) TUCHAROLL. This preprocessor produces the TUSUM file. The TUCHA file is input to this program. Of the four record types in TUCHA, only the ABF1 and the F2 record types are used. The layout of these records is given in Tables V-16 and V-17. Manual input is not required.

(3) POMCUSROLL. This preprocessor is used to produce the POMSUM file. The POMCUS file which is input to this program, is in a modified TUCHA transaction format. The layout of these records is given in Tables V-18 through V-20. The F3 detail record is not

TABLE V-16, TUCHA ABF1 Record Description

<u>Position</u>	<u>Type</u>	<u>Description</u>
1-5	A/N	Unit type code
9	N	Record type = 1
13-15	A	Unit level code
16	N	Deployment indicator code
17	A	Service code
18	A	Record security classification
19-33	A/N	Short type name
34	A/N	Invalid record indicator (T for TYPEA record)
35	A/N	Unit type status (C for cancelled)
36	A	F1 security classification
37-90	A/N	Type name
91-96	A/N	Originators UIC
97-102	N	Date of record creation
103-108	N	Date of last change
109-113	N	Authorized wartime personnel
114-118	N	Nonorganic passengers
119-121	N	Number or cargo categories
122-124	N	Count of F2 records
125	A/N	New F1 indicator
126-137	A/N	SRC
138	A	blank

TABLE V-17, TUCHA F2 Record Description

<u>Position</u>	<u>Type</u>	<u>Description</u>
1-5	A/N	Unit type code
6-7	A/N	Cargo category code
8	A	Containerization code
9	N	Record type = 2
13	A	F2 security classification
14	A/N	New F2 indicator
15	A	Heavy lift code
18-23	N	Square feet of cargo
24-35	N	Short tons of cargo (in tenths)
36-41	N	Measurement tons of cargo
42-47	N	Barrels of POL (in hundreds of barrels)
48-50	N	Number of F3 records required
51-53	N	Count of F3 records

TABLE V-18, POMCUS F1 Record

<u>Position</u>	<u>Type</u>	<u>Description</u>
7	N	Record type = 1
9-13	A/N	Unit type code
19-23	N	Nonorganic passengers
24-26	N	Number of F2 records
71-76	A/N	UIC of POMCUS unit
90-101	A/N	SRC
102-120	A/N	Not used

TABLE V-19, POMCUS F2 Record

<u>Position</u>	<u>Type</u>	<u>Description</u>
7	N	Record type = 2
9-13	A/N	Unit type code
14	A	POMCUS code A = TAT, B = Other exclusion, C = Prepo
34-35	A/N	Cargo category code
36	A	Containerization code
37-42	N	Square feet of cargo
45-50	N	Short tons of cargo (in tenths)
52-57	N	Measurement tons of cargo
59	A	Heavy lift code
60	N	Number of F3 records
71-76	N	UIC of POMCUS unit
90-101	A/N	SRC

TABLE V-20, POMCUS F3 Record

<u>Position</u>	<u>Type</u>	<u>Description</u>
7	N	Record type = 3
9-13	A/N	Unit type code
14	A	POMCUS code
19-20	A/N	Cargo category code
21	A	Containerization code
22-24	N	F3 identification number
25-38	A/N	Cargo description
39-42	N	Length
43-45	N	Width
46-48	N	Height
49-52	N	Square feet
53-55	N	Number of pieces

currently used but is documented here because other references are not available. Manual inputs are not required by POMSUM, however manual editing of the output may be required to ensure that the UIC identified on the Force file are present on the POMSUM file.

(4) MORSAROLL. The MORSAROLL preprocessor extracts other service movement requirements from the MORSA file. The layout of the MORSA file is shown in Table V-21. Manual input card formats are shown in Table V-22.

c. MRG Processors. Input to MRG programs is described in Tables V-23 through V-28. The Node Table is input to program GEO to assign SMOBSMOD nodes and default GELOC to each AREA used as an origin or destination. One entry as described in Table V-24 is required for each AREA. The SMOBSMOD node number will always be selected from this table. The default GELOC will be selected only if the actual LOCNA is not found on the GEO file.

d. Postprocessors. Program INLAY requires one card input as described in Table V-29, Program Build requires no manual input.

TABLE V-21, MORSA File Record Description

<u>Data name</u>	<u>Description a/</u>	<u>Start pos</u>	<u>No char</u>	<u>Type</u>
PLAN-ID	Identification code for plan	1	1	N
SERV	Service	2	1	A/N
SEQ	Sequence number of this record	3	4	N
RT	Record type	12	1	A/N
RD	Record description	13	1	A/N
REC-NAME	Record (unit or package description)	14	12	A/N
LEVEL		26	3	A/N
TMODE		30	1	A
AVAIL		32	3	N
RDD		35	3	N
NAME-O	Origin name	44	14	A/N
CODE-2-O	MORSA origin node	61	2	A/N
GEO-O	Origin GELOC	68	4	A
CODE-2-D	MORSA destination node	117	2	A/N
GEO-D	Destination GELOC	124	4	A
PAX		128	5	N
ACAMO		133	5	N
ASPLY		138	5	N
BULK		143	5	N
OVER-POL	(1) ST oversize cargo or, (2) thousand barrels POL (if RT=P)	148	5	N
OUTSZ		153	5	N
NAT		158	5	N
TOTAL-WT	Total cargo weight (less POL)	163	5	N
TPSN		168	5	N
SRC		173	11	A/N
FRN	Force requirements number	184	4	A/N
JCSTY		188	5	A/N
UIC		193	6	A/N
NSDA		217	5	N
Filler	N/A	222	57	A/N

a/ Where description is not given see Data Dictionary, this document.

TABLE V-22, MORSAROLL data

Card type 1 (one card for each origin code on MORSA file)

<u>Col</u>	<u>Data name</u>	<u>Type</u>	<u>Description</u>
1-2	CODE-2-0	A	MORSA origin node
4-6	ONODE	N	SMOBSMOD origin node

Card type 2 (one card for each destination code on MORSA file)

<u>Col</u>	<u>Data name</u>	<u>Type</u>	<u>Description</u>
1-2	CODE-2-D	A	MORSA destination node
4-6	DNODE	N	SMOBSMOD destination node

Card type 3 (one card only)

<u>Col</u>	<u>Data name</u>	<u>Type</u>	<u>Description</u>
1-2	DELTA-RDD	N	Enter number of days rollup desired for other services on RDD. For example, an entry of 10 will result in the following:

Actual RDD	RDD assigned by MORSAROLL
------------	---------------------------

0-9	5
10-19	15
20-29	25
etc	

2-4	DELTA-AVAIL	N	Enter number of days rollup desired for other services on AVAIL. For example, an entry of 05 will result in the following:
-----	-------------	---	--

Actual AVAIL	AVAIL assigned by MORSAROLL
--------------	-----------------------------

0-4	2
5-9	7
10-14	12
etc	

TABLE V-23, GEO Data

<u>Card type</u>	<u>Number required</u>	<u>Position</u>	<u>Type</u>	<u>Description</u>
1	1	1	N	Number of destination theaters (number of type 2 cards to follow--maximum 7)
2	1 per theater	1	N	Theater Code to select input force records
		3-5	N	Day available at origin for all except reserve above-the-line units.
		7-10	A	Destination GELOC
		12-14	N	Destination node number
		16-18	N	Day available at origin for reserve component above-the-line units
3	1	1-2	N	Number of exception cards (type 4) to follow (maximum of 20)
4	1 per exception	1-5	N	TPSN
		7-10	A	Destination GELOC
		12-14	N	Destination node number
		16-18	N	Day available
		20-22	N	Required Delivery Date
		24	A	TMODE (P-optional, A-air, S-sea, Z-not moved)
		26	N	Destination theater
		28-31	A	Origin GELOC
		33-35	N	Origin node number

TABLE V-24, Node Table

<u>Position</u>	<u>Type</u>	<u>Description</u>
1-3	AN	AREA (position 1 blank if numeric)
4-12	AN	LOCNA
13-16	A	GELOC
17-21	AN	ARLOC
22-24	N	SMOBSMOD node number

TABLE V-25, LOG Data

<u>Card type</u>	<u>Number required</u>	<u>Position</u>	<u>Type</u>	<u>Description</u>
1	1	1	N	Number of destination theaters
		3	N	Number of theater sets
		5-11	N	Theater code for each destination theater (maximum 7--blank if less than 7)
		13-19	N	Theater set for each destination theater
		27	N	Plan identification for each theater set
2	1	1-4	A	APOE GELOC-not used
		6-9	A	SPOE GELOC-not used
		11-16	N	Accompanying supply (dry and packaged POL) for non-POMCUS units lbs/man--decimal point required
		18-23	N	Accompanying ammunition for non-POMCUS units lbs/man--decimal point required
		25-30	N	Accompanying supply (dry) for POMCUS units lbs/man--decimal point required
		32-37	N	Accompanying ammunition for POMCUS units lbs/man--decimal point required
3	1	1-2	N	Number of SRC substitution cards (type 4) to follow (may be zero)
4	1 per substitution	1-11	AN	SRC less ALO (position 10) that appears in Force but not on TUCHA
		13-23	AN	TUCHA SRC that was substituted for Force SRC

TABLE V-26, UPMDFI Data

<u>Card</u>	<u>Position</u>	<u>Type</u>	<u>Description</u>
1	1-2	N	Number of SRC substitution cards to follow
2	1-11	AN	SRC less ALO (position 10) that appears in Force but not on TUCHA
	13-23	AN	TUCHA SRC to be substituted for Force SRC

TABLE V-27, NUR Data (continued on next page)

<u>Card</u>	<u>Position</u>	<u>Type</u>	<u>Description</u>
1	1	N	Number of destination theaters
	3	N	Number of theater sets
	5-11	N	Theater Code for each destination theater
	13-19	N	Theater set for each destination theater
	21-27	N	Plan identification for each theater set
2	1	N	Theater set number
	3-5	N	D-Day--day on which personnel casualties start
	7-0	N	RDAY--day on which theater resupply build up begins
	11-13	N	BLDUP--day on which build up objective is to be met
	15-17	N	C-Day--all units with RDD less than or equal to C-Day are considered in-country
	19-21	N	A-Day--all resupply packages with RDD greater than or equal to A-Day will be assigned TMODE=S (sea)
	23-25	N	Last day of resupply period 1
	27-29	N	Last day of resupply period 2
	31-33	N	Last day of resupply period 3
	35-37	N	Last day of resupply period 4
	39-41	N	Last day of resupply period 5
	43-45	N	Last day of resupply period 6
	47-49	N	Last day of resupply period 7
	50-51	N	Last day of resupply period 8 (last day of consumption)
	55-57	N	Level of supply (days) required at BLDUP--dry
	59-61	N	Days of ammo required at BLDUP
	63-65	N	Days of POL required at BLDUP
	67-69	N	Minimum level (days) of supply (dry) prior to RDAY
	71-73	N	Minimum days of ammo prior to RDAY day of resupply
	75-78	N	Minimum days of POL prior to RDAY day of resupply

TABLE V-27, NUR Data (continued on next page)

<u>Card</u>	<u>Position</u>	<u>Type</u>	<u>Description</u>
3	1	N	Theater set number
	3-11	N <u>a/</u>	Short tons of dry PWRS
	13-21	N <u>a/</u>	Short tons of ammo PWRS
	23-31	N <u>a/</u>	Bbls of POL PWRS
	33-36	N <u>a/</u>	Fraction of in-country strength needed for filler
	38-40	N	RDD for 1st filler increment
	42-44	N	RDD for 2nd filler increment
	46-48	N	RDD for 3rd filler increment
	50-53	N <u>a/</u>	Fraction of total fillers in 1st increment
	55-58	N <u>a/</u>	Fraction of total fillers in 2nd increment
	60-63	N <u>a/</u>	Fraction of total fillers in 3rd increment
	65-68	A	Destination GEO code
	70-72	N	Destination node number
4	1	N	Theater set number
	3-7	N <u>a/</u>	Replacements/1000/day 1st period
	9-13	N <u>a/</u>	Replacements/1000/day 2nd period
	15-19	N <u>a/</u>	Replacements/1000/day 3rd period
	21-25	N <u>a/</u>	Replacements/1000/day 4th period
	27-31	N <u>a/</u>	Replacements/1000/day 5th period
	33-37	N <u>a/</u>	Replacements/1000/day 6th period
	39-43	N <u>a/</u>	Replacements/1000/day 7th period
5	45-49	N <u>a/</u>	Replacements/1000/day 8th period
	1	N	Theater set number
	3-8	N <u>a/</u>	Dry accompanying supply for non-POMCUS units lbs/man
	10-15	N <u>a/</u>	Accompanying ammo for non-POMCUS units lbs/man
	17-22	N <u>a/</u>	Accompanying POL for non-POMCUS units lbs/man
	24-29	N <u>a/</u>	Dry accompany supply for POMCUS units lbs/man
	31-36	N <u>a/</u>	Accompanying ammo for POMCUS units lbs/man
	38-43	N <u>a/</u>	Accompanying POL for POMCUS units lbs/man

a/Decimal point required

TABLE V-27, NUR Data (concluded)

<u>Card</u>	<u>Position</u>	<u>Type</u>	<u>Description</u>
6	1	N	Theater set number
	2	N	Resupply type (1-dry, 2-ammo, 3-POL)
	4-9	N <u>a/</u>	lbs/man/day 1st period
	11-16	N <u>a/</u>	lbs/man/day 2nd period
	17-22	N <u>a/</u>	lbs/man/day 3rd period
	24-29	N <u>a/</u>	lbs/man/day 4th period
	31-36	N <u>a/</u>	lbs/man/day 5th period
	38-43	N <u>a/</u>	lbs/man/day 6th period
	45-51	N <u>a/</u>	lbs/man/day 7th period
	53-58	N <u>a/</u>	lbs/man/day 8th period
7 <u>b/</u>	1-4	A	Origin GELOC
	6-8	N	Origin node number
	10-13	N <u>a/</u>	Fraction of dry from this origin
	15-18	N <u>a/</u>	Fraction of ammo from this origin

a/Decimal point required

b/20 type 7 cards are required-use blank if necessary-last two specify replacement and POL origins

TABLE V-28, PDKAGE Data

<u>Card</u>	<u>Position</u>	<u>Type</u>	<u>Description</u>
1	1-2	N	Packaging interval on RDD
	4-5	N	Packaging interval on availability
2	1	N	Theater number
	3-5	N	RDD to begin requiring all units to theater by mixed mode
	7	N	Theater number
	9-11	N	RDD to start all mixed mode
	13	N	Theater number
	15-17	N	RDD to start all mixed mode
	19	N	Theater number
	21-23	N	RDD to start all mixed mode
	25	N	Theater number
	27-29	N	RDD
	31	N	Theater number
	33-35	N	RDD
	37	N	Theater number
	39-41	N	RDD

TABLE V-29, INLAY Data

<u>Position</u>	<u>Data entry</u>
1	FICOD for NUR records
3-8	EDATE for NUR records

COMPUTER ASSISTED MATCH PROGRAM
(CAMP)

CHAPTER VI
JOB UTILIZATION

1. Force Match Algorithm. This section describes the output from FMA. The ultimate output is the FAS tape overlaid with match results. However, intermediate reports are produced that should be examined. Examples are shown in Chapter VIII. To analyze the final output appropriately, it is necessary to load the FAS tape into the UDS system using Program BUILD. Detailed and summary reports can then be obtained and analyzed.

a. Program SORTUDS. This program provides a listing of the force in COMPO/UIC sequence. Data elements displayed are identified in Table VI-1.

b. Program FSORT provides a copy of the manual entry data and a listing of each of the five files produced. Each listing has the same format as described in Table VI-2. The Match Candidate file is sorted in match sequence; all other files are in the original COMPO/UIC sequence.

c. Program RSORT displays the manual entry data, a count of the units, and two reports: the UIN report (deleted units) and the REQ report (requirements) in match sequence. The reports are described in Table VI-3.

d. Program ATL prints the manual entry tables, the input data file (ATL) and the output data file (MATL). The output file is exactly the same as the input file (Table VI-2) except that the ADC02 and ADC03 fields have been overlaid with Time Period and RDD, respectively.

e. Program ALT prints the manual entry table and the output file (MALT) that is exactly the same as the input file (Table VI-2) except that the ADCON field has been overlaid with Destination Theater and RDD.

f. Program MATCH produces a report (Table VI-4) that readily identifies matches, overages, and shortfalls. The first two data items displayed are the REQ-SRC (FASTALS requirement SRC) and FORCE-SRC (FAS file SRC). Entries in both columns indicate a match. An entry in FORCE-SRC and a blank REQ-SRC indicates an overage. An

TABLE VI-1, SORTUDS Report

<u>Column</u>	<u>Data element</u>
1-6	UIC
9-13	TPSN
16	COMPO
19-21	FPLAN
24-26	ROBCO
29-40	SRC
43-47	JCS Type Unit Code
50-51	Level
55-56	CARRS
29-62	Unit number
65-66	Branch
69-86	Unit description
99-107	Home station
109-111	LOCCO
113-114	DSCMP
117	ADC01
119-120	ADC02
122-124	ADC03

TABLE VI-2, FSORT Reports

<u>Column</u>	<u>Data element</u>
1-7	SRC (positions 1-5, 8,9)
8	COMPO
9	MCODE
11-15	DAMPL
16-21	UIC
22-26	SRC (positions 6,7, 10-12)
27-28	Branch
29-31	Level
32-49	Unit description
50-54	TPSN
55-57	ROBCO
58-66	Home station
67-69	LOCCO
70-74	STAGR
75-80	ADCON
81-86	Sequence number
87-91	FSORT Key a/

a/FSORT Key has the following meaning:

```

Lock Out file: Reason for lock out
ATL file:      YES or NO to indicate ADC03 set based on manual
               input to FSORT
Other files:   blank

```


TABLE VI-3, RSORT Report

<u>Column</u>	<u>Data element</u>
1-5	SRC (truncated)
7-8	SRC (positions 8,9)
11	Time period
13-14	SRC (positions 6,7)
16	SRC (position 10)
18	SRC (position 12)
20	SRC (position 11)
22-24	Level
26-46	Unit description
47-50	Rules
52	Logical Region of Employment
54	Combining rule
56	Rounding rule
58-62	TPSN <u>a/</u>
64-68	STOFF <u>a/</u>
70-74	STWOF <u>a/</u>
76-80	STENL <u>a/</u>
82-86	STAGR <u>a/</u>
88-90	FASTALS strength
92-93	Branch
95-97	UIN

a/Used only in the generation of notional units.

TABLE VI-4, MATCH Report

<u>Column</u>	<u>Data element</u>
1-8	REQ-SRC
11-18	FORCE-SRC
23-24	Branch
28-30	Level
33-50	Unit description
54	COMPO
58-63	UIC
65-69	TPSN
71-73	UIN
76	Time Period
79-82	Requirement sequence number
84	MCODE
88-90	ROBCO
93-101	Home station
104-106	LOCCO
109-113	FASTALS strength
115-119	FAS strength ^{a/}
121-125	Force sequence number

^{a/}For notional units, this is the structured aggregated strength from the TOE file.

entry in REQ-SRC and a blank in FORCE-SRC indicates a shortfall for which a notional unit has been generated. This report is prepared in SRC sequence to facilitate analysis. The Match Report is the key report of the match process as it identifies the required force structure (except for hand played units) and the status of the actual force in relation to the requirements.

g. Programs LAYIN and BINCOPY do not produce any significant printed output. However, the FAS tape produced by BINCOPY can be loaded into a keyed access UDS file using program BUILD. UDS provides an enhanced capability for analysis of the match results. Further, the file produced by LAYIN (FOUT) is used as input to MRG.

2. Movement Requirements Generator. In this section the output from MRG is described. The ultimate output, will be movement requirements in SMOBSMOD and/or MORSA format. However, intermediate output is available at several points in the processing to assist the user in analysis. Examples of output are shown in Chapter VIII.

a. Preprocessors. As stated in Chapter III, MRG includes four preliminary processors. Each one produces an intermediate file which is used in subsequent processing and a printed report that should be perused by the user. The use of these four reports is described in the following paragraphs.

(1) TUCHAROLL. The output from TUCHAROLL is a summary of movement characteristics of each type unit. The report is printed in SRC sequence and includes two lines per type unit. The data elements are described in Table VI-5. Under each cargo category, the first line displays weights in tenths of short tons, and the second line displays square feet of cargo if appropriate. Occasionally, the user may discover errors in the TUCHA file. The correct data can usually be ascertained through coordination with ODCSLOG, and the intermediate mass storage file produced by TUCHAROLL, in the same format as the printed report, can be corrected using the edit processor. The output from TUCHAROLL is also useful in determining appropriate SRC substitutions. If the force being processed contains SRC not on the TUCHA file, substitutions must be made subsequent to running program LOG.

(2) POMCUSROLL. The output from POMCUSROLL is similar to that from TUCHAROLL except that the report is sorted in UIC sequence and four lines are printed for each unit. The first two lines indicate quantities to be moved and correspond to the first two lines in the TUCHAROLL report. The next two lines indicate quantities prepositioned. The POMCUS report should be checked against the force to ascertain if all POMCUS units are represented. If not, substitutions can be made using the edit processor, or a new POMCUS file can be obtained from ODCSLOG.

TABLE VI-5, Data Elements Displayed by TUCAROLL

<u>Data element</u>		<u>Description</u>
SRC		Standard requirements code, line number
ULC/UTC	line 1: line 2:	Unit level code/unit type code Unit description
PAX	line 1: line 2:	Number passengers Unit strength
A0		Vehicles, non-air transportable (NAT)
B0		Non-self-deployable aircraft (NSDA) NAT
B1		NSDA outsize
A1		Vehicles, outsize
J1		Nonvehicular outsize cargo
B2C		NSDA oversize, containerizable
A2C		Vehicles, oversize, containerizable
A2D		Vehicles, oversize, noncontainerizable
J2D		Nonvehicular cargo, oversize, noncontainerizable
J2C		Nonvehicular cargo, oversize, containerizable
A3C		Vehicles, Bulk, containerizable
J3C		Nonvehicular cargo, bulk, containerizable
J7C		Accompanying dry bulk supplies
AMMO		Accompanying ammunition
UE/B	line 1: line 2:	Total weight of unit equipment Total square feet of NSDA
AS/A	line 1: line 2:	Total accompanying supply weight _a / Total square feet of vehicles

a/Not used in MRG

(3) MORSAROLL. The reports are produced by MORSAROLL. The first is a listing of the SMOBSOD input data (A and B cards only, the C card is produced but not listed) for other service movements. The A, B, and C cards are described in Reference 2. The report is listed in package number sequence as derived by sorting the other service movement data on RDD, availability date, travel mode, origin node, and destination node. The format is exactly the same as that for SMOBSMOD input. The A card shows package number, RDD, and availability date. One B card is produced for each cargo category included in the package. Current categories are listed at Table VI-6. Each B card displays availability date, destination node, origin node, travel mode, number of passengers (thousands) cargo quantity thousands of barrels of POL or thousands of short tons for other cargo) and the cargo type. The second report is a detailed listing of the units making up each package. The data elements on this report are self-explanatory except for OV/POL. Normally all cargo quantities are input from MORSA in short tons, and the OV/POL field displays quantity of oversize cargo. However, if the cargo is POL (record type 'P') then the OV/POL field displays barrels in thousands. This program also provides a listing of all records for which the origin or destination could not be found in the node tables. A code (1 for origin, 9 for destination) indicates which type node was not located. Also displayed are the MORSA node identification, total weight, bulk, number of passengers, RDD, and availability date. Although the program does not error terminate when a node is not found, the user should update the node tables and assign a SMOBSMOD node to each MORSA node.

(4) ARLOCSORT. Two reports are provided by ARLOCSORT. The first report displays the entire Army Location file in Army Location code (ARLOC) sequence and in abbreviated location name (LOCNA) sequence. The data elements displayed have been listed in Table V-15. The second report is sorted by AREA and displays only AREA, LOCNA, GELOC and ARLOC. These are the same data elements contained in the GEO Location file and subsequently used by Program GEO.

b. Main processors. Each of the MRG programs provides output to assist the user in analysis of results.

(1) GEO. All manual entry data are formatted and displayed to the user. The user should carefully check the program output to verify the accuracy of manual entry data. Also displayed is the number of records in the GEO file and Node table. The GEO file is the same file output by ARLOCSORT, so the number of records should agree. The Node table is created from card input, so the number of

records can easily be checked. The last direct output from GEO identifies the number of locations on the Force file that could not be found on the GEO file and the average number of accesses to the GEO file per unit. Locations not found on the GEO file are assigned a notional GELOC from the Node table based on LOCCO or MBLOC. If the LOCCO and MBLOC values are not found in the Node table, the program will terminate abnormally after identifying all such values. The user should then update the Node table. GEO also produces a Movement Data file 1 (MDF1) which can be listed or edited separately by the user. This file displays the data elements shown in Table VI-7.

(2) LOG. This program may terminate abnormally if there are discrepancies in the input data. The user must correct the discrepancies and rerun the program in the event of abnormal termination of the program. The next paragraphs describe the output under abnormal and normal conditions.

(a) Abnormal termination. If any SRC appearing on the MDF1 file are not found on the TUSUM file, these SRC will be listed. The user must then determine appropriate substitute SRC whose movement characteristics will be used in lieu of the SRC that could not be found. Programs UPMDF1 and LOG are then rerun with the list of substitute SRC as input.

(b) Normal termination. All manual entry data are formatted and displayed to the user who should carefully check this output to verify the accuracy of the manual entry data. The number of units in each theater set is displayed. A list of units, that are identified as POMCUS units on the Force file but that are not located on the POMCUS file, is then printed. Such mismatches do not cause program termination because these units will be assigned zero deployment weight. However, the user should peruse the list to determine if an error exists on one of the files. Program LOG also produces a file of MORSA cards for each unit. This file can be printed or edited as desired by the user.

(3) UPMDF1. This program does not produce any user output.

(4) NUR. Manual entry data are formatted and displayed for user perusal. This output should be carefully checked to assure correctness. NUR prints a supply status summary in 5-day increments for each theater. Data elements are listed in Table VI-8. NUR also identifies units on MDF1 for which no corresponding Theater Code was input to the program. No resupply or replacement quantities are generated for these units. If the user wants these quantities determined, appropriate manual entry data must be provided to NUR. NUR also produces a file of MORSA cards for the nonunit movement requirements. This file can be listed or edited by the user as desired.

TABLE VI-6, SMOBSMOD Cargo Categories

<u>Category</u>	<u>Cargo type</u>
1	PAX
2	VEH-NAT
3	NSDA-NAT
4	NSDA-OUT
5	VEH-OUT
6	POL
7	NV-OUT
8	NSDA-OVER-C
9	NSDA-OVER-NC
10	VEH-OVER-C
11	VEH-OVER-NC
12	NV-OVER-NC
13	NV-OVER-C
14	VEH-BULK-C
15	NV-BULK-C
16	DRY RESUPPLY
17	AMMO

TABLE VI-7, MDF1 Data Elements

<u>Data element</u>	<u>Description</u>
SRC	Standard requirement code
UIC	Unit identification number
TPSN	Troop program sequence number
ORIGIN	Origin GELOC
Destination	Destination GELOC
Origin-node	SMOBSMOD node number
Destination-node	SMOBSMOD node number
RDD	Required delivery date
AVAIL	Availability date
MODE	MORSA travel mode
Theater code	See Data Dictionary (this document)
MCODE	See Data Dictionary (this document)
Unit description	Description of the type unit

TABLE VI-8, Supply Status Summary

<u>Data element</u>	<u>Description</u>
DAY	Effective day of summary
THR	Theater code
UNITS	Number of units deployed to the theater (cumulative)
STRENGTH	Authorized theater strength used to determine number of casualties
REPLACEMENTS	Number of replacements required
TYPE	'1' = Dry bulk cargo '2' = Ammunition '3' = POL
DAYS OH	Number of days of supply on hand in theater stocks (by type)
DAYS DESRD	Number of days of supply required to be in theater stocks (by type)
TONS OH	Number of short tons on hand
TONS REQD	Shortfall in theater stocks (POL units and thousands of barrels)

(5) MERGEMORSA. A user output is not produced by this program.

(6) PCKAGE and PRTPACK. The printed output is self-explanatory. Two files are also produced that can be listed or edited by the user. One file contains SMOBSMOD input (A, B, and C cards) as defined in paragraph 2a(3) of this chapter. The second file contains a detail listing of units (identified by UIC and TPSN) within each Army package.

c. Post processors. MRG includes two programs designed to interface with UDS. Use of these programs is optional with the user but is strongly recommended because of the versatile report generation capability of UDS. The UDS reports provide a valuable aid to the user in analysis of MRG results. Two programs are required to produce a UDS file. Program INLAY overlays movement data on selected fields of the FAS file, creates additional FAS records for nonunit movements, and produces a modified FAS tape. Program BUILDMRG accepts the modified FAS tape and produces a UDS file with the new data. Table VI-9 identifies the MRG data items overlaid on the FAS file and the UDS data names associated with the overlaid data. A special version of the UDS report generator (39FACT.MAINABS) should be used to extract reports from this file. See Reference 1 for an explanation of UDS.

TABLE VI-9, MRG Data Items Overlaid on FAS File

MRG data	Size	FAS position	UDS name
Availability Date	3	34-36	AVAL1
Record Type	1	37	RTCCC
Record Description	1	38	RDCCC
Sequence Number	4	48-53	CCNUM
Origin Geographic Location Code	4	54-57	ORGEO
Destination Geographic Location Code	4	58-61	DEGEO
Travel Mode	1	62	TMODE
Required Delivery Date	3	110-112	REQDA
Destination node <u>a/</u>	3	113-115	DNODE
Origin node <u>a/</u>	3	116-118	ONODE
Package Number <u>a/</u>	4	163-167	PKG NR
Unit level <u>b/</u>	3	204-206	} UNTDS
Unit description <u>b/</u>	12	207-224	
Bulk (short tons)	6	228-232	BULK1
Oversize (short tons)	6	233-237	OVRSZ
Outsize (short tons)	6	238-242	OUTSZ
NAT (short tons)	6	243-247	NATAA
NSDA (square feet)	6	248-252	NSDA1
Total weight (short tons)	6	253-257	DEPST
Accompanying ammunition (short tons)	6	258-262	ACAMO
POL (000 barrels)	6	263-267	POLBB
Passengers	6	268-272	PAXAA
Accompanying dry cargo (short tons)	6	273-277	ASPLY

a/Data items obtained from SMOBSMOD input. All other data items obtained from MORSA input.

b/Overlaid for nonunit records only.

APPENDIX A
STUDY CONTRIBUTORS

PRECEDING PAGE BLANK-NOT FILMED

COMPUTER ASSISTED MATCH PROCESS (CAMP)

APPENDIX A
STUDY CONTRIBUTORS

1. Study Director

Mr Gary L. Martin, Joint and Strategic Forces Directorate

2. Study Team

Ms Patricia Fleming, Methodology and Resources Directorate
Mr Ivan Keller, Force Concepts and Design Directorate
MAJ Ernest R. Montagne, Jr, Joint and Strategic Forces Directorate
Mr Rene Plourde, Methodology and Resources Directorate
Ms Avonale Stephenson, Project Planning and Control Office

3. Support Personnel

Ms Betty Spence
Ms Armin Marvin
Ms Vivian Layman

APPENDIX B
REFERENCES

PRECEDING PAGE BLANK-NOT FILMED

COMPUTER ASSISTED MATCH PROGRAM (CAMP)

APPENDIX B
REFERENCES

1. US Army Concepts Analysis Agency, "Capabilities and Use of the CAA Unit Data System (UDS)," Draft Documentation, Bethesda, MD, 23 Jul 74.
2. _____, "Strategic Mobility Simulation Model (SMOBSMOD) User's Manual," Draft Documentation, Bethesda, MD, 30 May 74.
3. Defense Communications Agency, "Movement Requirements for Staff Planning and Special Studies Applications (MORSA)," User's Manual, Logistic Data Branch (ATTN: C318), Command and Control Technical Center. To be published.
4. Dept of Army, "Force Accounting System (FAS)," User's Guide, Washington, DC, Oct 72.
5. Research Analysis Corporation, "The FOREWON Force-Planning System," App C of "User's Guide for Force Analysis Simulation of Theater Administrative and Logistic Support (FASTALS)," RAC-R-86, May 71.
6. Dept of Army, "Army Location Codes, States Within the United States," DA Circular 525-10-3, Washington, DC, 30 Aug 74.
7. _____, "Army Location Codes, Foreign Locations," DA Circular 525-10-4, Washington, DC, 30 Aug 74.
8. Defense Communications Agency, "Type Unit Data File (TUCHA)," in Users Manual of Computer System Manual, CSM UM 157-74, Jul 74.
9. Dept of Army, "Troop Program Sequence Number," AR 18-19, Washington, DC, 16 Aug 72.

COMPUTER ASSISTED MATCH PROGRAM (CAMP)

APPENDIX C
GLOSSARY

PRECEDING PAGE BLANK-NOT FILMED

COMPUTER ASSISTED MATCH PROGRAM (CAMP)

APPENDIX C Glossary

<u>Term</u>	<u>Definition</u>
ALT	A computer program. A file output by program FSORT for input to program ALT
ammo	ammunition
ATL	Above-the-line. Refers to major combat forces (divisions and separate brigades). All units with TPSN first position 0 or 1 are above-the-line. A computer program. A computer file output by program FSORT for input to program ATL
BINCOPY	A computer program
BUILD	A computer program
BLDUP	day stockage objective is to be achieved
BUILDMRG	A computer program
CAA	Concepts Analysis Agency
CAMP	Computer Assisted Match Program
CAMPSUM	A program RSORT input computer file
CAN	A program Match data input file. A program FSORT output file. The match candidate file
CCTC	Command and Control Technical Center
COBOL	common business oriented language
DA	department of the Army
DC	display and compute
DO	display only

<u>Term</u>	<u>Definition</u>
FAS	Force Accounting System
FASTALS	Force Analysis Simulation of Theater Administration and Logistic Support
FMA	force match algorithm
FORTRAN	Formula Translation (a scientific programming language for computers)
FSORT	A computer program
GEO	A computer program that produces the Movement Data File 1
HPU	Hand played units. A computer file output by program FSORT for input to program LAYIN
INLAY	A computer program
JCS	Joint Chiefs of Staff
K	thousand
LAYIN	A computer program
LOG	A computer program
LOU	Locked out units. Units not considered for matching. A computer file output by program FSORT for input to program LAYIN
MALT	A program LAYIN computer input file. Also a program ALT output file
MATCH	A computer program
MATL	A file output by program ATL for input to program LAYIN
MERGEMORSA	A computer program
MORSA	Mobility Requirements for Staff Analysis
MORSAROLL	A computer program that provides preliminary processing for data base building

<u>Term</u>	<u>Definition</u>
MRD	Methodology and Resources Directorate, CAA
MREQ	A computer file output by program MATCH for input to program LAYIN (includes matches, overages, and notional units)
NUR	A computer program
ODCSOPS	Office, Deputy Chief of Staff for Operations and Plans
PCKAGE	A computer program
POL	petroleum, oils and lubricants
POMCUS	prepositioning of materiel configured to unit sets. A computer file input to MRG
POMCUSROLL	A computer program that provides preliminary processing for data base building
POMSUM	A summary file produced by POMCUSROLL
pos	position
PRTPCK	A computer program
PWRS	prepositioned war reserve stock
RDAY	day theater resupply build up begins
RDD	required delivery date
REQ	A computer file output by program RSORT for input to program MATCH
ROLLUP-MODIFIED	A new version of CAMPSUM
RSORT	A computer program
SORTARLOC	A computer program
SORTUDS	A computer program that provides preliminary processing for data base building
TOE	table(s) of organization and equipment

<u>Term</u>	<u>Definition</u>
TOE/TPSN	table(s) of organization and equipment/Troop Program Sequence Number. A computer file
TPSN	Troop Program Sequence Number (see reference 9)
TUCHA	Type unit data, a computer file input to MRG
TUCHAROLL	A computer program that provides preliminary processing for data base building
TUSUM	A summary file produced by TUCHAROLL
UDS	Unit Data System. Also, a computer file
UPMDF1	A computer program
WWMCCS	worldwide military command and control system
66 ROLLOUT	A third version of CAMPSUM (used if movement requirements are to be generated)
66 ROLLUP	The principal version of Program CAMPSUM

BEST AVAILABLE COPY

COMPUTER ASSISTED MATCH PROGRAM (CAMP)

CHAPTER VII

PROGRAM LISTINGS

BEST AVAILABLE COPY


```
WELT-L 24MONTAGNE-BUILD
ELT007 RL1870 08/10-07:40:07-(3.)
000001 000 010010 IDENTIFICATION DIVISION.
000002 000 010020 PROGRAM-ID. 'BUILD'.
000003 000 010030 AUTHOR. J MEYEROWITZ.
000004 000 010040 DATE WRITTEN. 02 MAY 1973.
000005 000 010060 REMARKS. READ A FAS FILE FROM TAPE
000006 000 010070 307 CHARACTERS BLOCKED 24 PER RECORD.
000007 000 010080 WRITE A UDF FILE TO MASS STORAGE USING ISFM
000008 000 010090 360 CHARACTERS PER RECORD.
000009 000 020010 ENVIRONMENT DIVISION.
000010 000 020020 CONFIGURATION SECTION.
000011 000 020030 SOURCE-COMPUTER. UNIVAC-1108.
000012 000 020040 OBJECT-COMPUTER. UNIVAC-1108.
000013 000 020050 INPUT-OUTPUT SECTION.
000014 000 020060 FILE-CONTROL.
000015 000 020070 SELECT MASTER-FILE ASSIGN TO MASS-STORAGE FORCE
000016 000 020080 ACCESS MODE IS RANDOM SYMBOLIC KEY IS REAL-KEY
000017 000 020085 ORGANIZATION IS INDEXED FILE-DESCRIPTION IS FILE-DESCRIPT.
000018 000 020120 SELECT FAS-FILE ASSIGN TO UNISERVU FAS-FILE.
000019 000 020130 SELECT PRINT-FILE ASSIGN TO PRINTER.
000020 000 030010 DATA DIVISION.
000021 000 030020 FILE SECTION.
000022 000 030030 FD MASTER-FILE
000023 000 030040 LABEL RECORDS ARE STANDARD
000024 000 030050 DATA RECORD IS MASTER-REC. PICTURE IS X(10752).
000025 000 030060 01 MASTER-REC
000026 000 031010 FD FAS-FILE
000027 000 031020 LABEL RECORD IS OMITTED
000028 000 031030 RECORD CONTAINS 7368 CHARACTERS
000029 000 031035 RECORDING MODE IS 1
000030 000 031040 DATA RECORD IS TAPE-BLOCK. PICTURE IS X(7368).
000031 000 031050 01 TAPE-BLOCK
000032 000 032010 FD PRINT-FILE
000033 000 032020 RECORD CONTAINS 132 CHARACTERS
000034 000 032030 LABEL RECORD IS OMITTED
000035 000 032040 DATA RECORD IS PRINT-LINE. PICTURE IS X(132).
000036 000 032050 01 PRINT-LINE
000037 000 040010 WORKING-STORAGE SECTION.
000038 000 040020 77 NDEX PICTURE IS 99 COMP SYNC RIGHT VALUE IS 0.
000039 000 040030 77 N PICTURE IS H9 VALUE IS ZERO.
000040 000 040100 01 WORK-BLOCK.
000041 000 040110 02 WORK-REC OCCURS 24 TIMES.
000042 000 040120 03 PART-1.
000043 000 040130 04 FILLER PICTURE IS X.
000044 000 040140 04 WORKID.
000045 000 040143 05 FICOD PICTURE IS X.
000046 000 040144 05 COMPO PICTURE IS X.
000047 000 040145 05 UICCC PICTURE IS X(6).
000048 000 040146 05 EDATE PICTURE IS X(6).
000049 000 040150 04 FILLER PICTURE IS X(11).
000050 000 040160 03 PART-2.
000051 000 04162 04 FILLER PIC X(12).
000052 000 04 THEATER-CODE PIC X.
000053 001 04 FILLER PIC X(31).
000054 001 04 DSCMP PIC XX.
000055 001 04 FILLER PIC XXX.
```

FMA PRE-PROCESSOR PROGRAM BUILD ••UNCLASSIFIED••

```
000056 000 04164 04 ESCON PIC X*
000057 000 04166 04 FILLER PIC X(108)*
000058 000 04166 04 TRCON PIC X*
000059 003 04166 04 FILLER PIC X(122)*
000060 000 050010 01 DATA-REC*
000061 000 050020 02 SEG-1 PICTURE IS X(26)* VALUE IS 'X'
000062 000 050030 02 FILLER PICTURE IS X
000063 000 050040 02 SEG-2 PICTURE IS X(333)*
000064 000 060010 01 PRT-AREA*
000065 000 060020 02 PRT-ITEM PICTURE IS X(16) OCCURS 7 TIMES*
000066 000 080010 01 ER-LINE*
000067 000 080020 02 FILLER PICTURE IS X(12) VALUE IS 'INVALID KEY '
000068 000 080030 02 ER-KEY PICTURE IS X(14)*
000069 000 150010 01 REAL-KEY*
000070 000 150020 02 REC-KEY PICTURE IS X(14)*
000071 000 150030 02 FILLER PICTURE IS X(4)*
000072 000 180010 01 FILE-DESCRIPT*
000073 000 180020 02 NUMBER-OF-RECORDS PICTURE IS M9(10) VALUE IS 12000*
000074 000 180030 02 RECORD-LENGTH PICTURE IS M9(10) VALUE IS 360*
000075 000 180040 02 RECORD-LENGTH-MAX PICTURE IS M9(10) VALUE IS 0*
000076 000 180050 02 RECORD-KEY-LENGTH PICTURE IS M9(10) VALUE IS 18*
000077 000 180060 02 OVERFLOW-AREA PICTURE IS M9(10) VALUE IS 10*
000078 000 180070 02 I-O-F PICTURE IS M9(10) VALUE IS 1000*
000079 000 180080 02 DATA-NAME-8*
000080 000 180090 03 DATA-NAME-9 PICTURE IS X(3)*
000081 000 180100 03 DATA-NAME-10 PICTURE IS X(3)*
000082 000 190010 01 INFORM*
000083 000 190020 02 NUMBER-OF-BLOCKS PICTURE IS M9(10)*
000084 000 190030 02 NUMBER-OF-INDEX-BLOCKS PICTURE IS M9(10)*
000085 000 190040 02 NUMBER-OF-OVERFL-BLOCKS PICTURE IS M9(10)*
000086 000 190050 02 NUMBER-OF-RECORDS PICTURE IS M9(10)*
000087 000 190060 02 NUMBER-OF-RECORDS-IN-10F PICTURE IS M9(10)*
000088 000 190070 02 NUMBER-OF-RECORDS-DELETED PICTURE IS M9(10)*
000089 000 190080 02 NUMBER-OF-RECORDS-READ PICTURE IS M9(10)*
000090 000 190090 02 NUMBER-OF-RECORDS-READ-FROM-IF PICTURE IS M9(10)*
000091 000 190100 02 NUMBER-OF-RECORDS-WRITTEN PICTURE IS M9(10)*
000092 000 195010 01 INFORM-FL-DATA*
000093 000 195020 02 NUMBER-OF-BLOCKS PICTURE IS 9(6)*
000094 000 195030 02 NUMBER-OF-INDEX-BLOCKS PICTURE IS 9(6)*
000095 000 195040 02 NUMBER-OF-OVERFL-BLOCKS PICTURE IS 9(6)*
000096 000 195050 02 NUMBER-OF-RECORDS PICTURE IS 9(6)*
000097 000 195060 02 NUMBER-OF-RECORDS-IN-10F PICTURE IS 9(6)*
000098 000 195070 02 NUMBER-OF-RECORDS-DELETED PICTURE IS 9(6)*
000099 000 195080 02 NUMBER-OF-RECORDS-READ PICTURE IS 9(6)*
000100 000 195090 02 NUMBER-OF-RECORDS-READ-FROM-IF PICTURE IS 9(6)*
000101 000 195100 02 NUMBER-OF-RECORDS-WRITTEN PICTURE IS 9(6)*
000102 000 200000 PROCEDURE DIVISION*
000103 000 200010 START-TEST*
000104 000 200020 OPEN INPUT FAS-FILE WITH NO REWIND*
000105 000 200030 OPEN OUTPUT PRINT-FILE*
000106 000 200040 OPEN OUTPUT MASTER-FILE*
000107 000 210000 TAPE-READ*
000108 000 210010 MOVE ZERO TO NDEX*
000109 000 210020 READ FAS-FILE AT END GO TO FINI*
000110 000 210030 MOVE TAPE-BLOCK TO WORK-BLOCK*
000111 000 220000 GET-ITEM*
000112 000 220010 ADD 1 TO NDEX*
```

FMA PRE-PROCESSOR PROGRAM BUILD ••UNCLASSIFIED••

```

000113 000 220020 IF NDEX = 25 GO TO TAPE-HEAD.
000114 000 220021 IF WORKID (INDEX) = SPACES GO TO GET-ITEM.
000115 000 220025 IF WORKID (INDEX) = '#####' GO TO GET-ITEM.
000116 000 220030 MOVE THEATER-CODE (INDEX) TO TRCON (INDEX).
000117 000 220030 MOVE PART-1 (INDEX) TO SEG-1.
000118 000 220040 MOVE PART-2 (INDEX) TO SEG-2.
000119 000 220050 MOVE WORKID (INDEX) TO REAL-KEY.
000120 000 220055 ADD 1 TO N.
000121 000 220060 IF N = 8 PERFORM LINE-LIST THRU X-LIST.
000122 000 220065 WRITE MASTER-REC FROM DATA-REC INVALID KEY GO TO ERR-0.
000123 000 220070 MOVE REAL-KEY TO PRT-ITEM (N).
000124 000 220080 GO TO GET-ITEM.
000125 000 225010 LINE-LIST.
000126 000 225020 MOVE 1 TO N.
000127 000 225030 WRITE PRINT-LINE FROM PRT-AREA.
000128 000 225040 MOVE SPACES TO PRT-AREA.
000129 000 225050 X-LIST.
000130 000 225060 EXIT.
000131 000 226010 ERR-0.
000132 000 226020 MOVE REAL-KEY TO ER-KEY.
000133 000 226025 WRITE PRINT-LINE FROM ER-LINE.
000134 000 226030 GO TO FINI.
000135 000 230000 FINI.
000136 000 230010 CLOSE FAS-FILE.
000137 000 230020 CLOSE MASTER-FILE USING INFORM.
000138 000 230030 MOVE CORRESPONDING INFORM TO INFORM-FL-DATA.
000139 000 230035 WRITE PRINT-LINE FROM PRT-AREA.
000140 000 230040 WRITE PRINT-LINE FROM INFORM-FL-DATA AFTER 2 LINES.
000141 000 230050 CLOSE PRINT-FILE.
000142 000 230060 STOP RUN.

```

END ELT.

@HDG FMA/MRG PRE-PROCESSOR PROGRAM SORTUOS **UNCLASSIFIED** *L10


```
WELT,L 32MARTIN.SORTUDS2FAS
ELT007 RL1870 08/16-07:40:09-11.1)
000001 000 IDENTIFICATION DIVISION. SORTUDS2FAS
000002 001 PROGRAM-ID.
000003 000 ENVIRONMENT DIVISION.
000004 000 CONFIGURATION SECTION.
000005 000 SOURCE=COMPUTER. UNIVAC-1108.
000006 000 OBJECT=COMPUTER. UNIVAC-1108.
000007 000 INPUT-OUTPUT SECTION.
000008 000 FILE-CONTROL.
000009 000 SELECT PRNTFL ASSIGN TO PRINTER.
000010 000 SELECT CARDFL ASSIGN TO CARD-READER.
000011 000 SELECT UD5FL ASSIGN TO UNISERVO SRC-TAPE.
000012 000 SELECT SRTFL ASSIGN TO UNISERVO UIC-TAPE.
000013 000 SELECT SRTFL ASSIGN TO MASS-STORAGE XA.
000014 000 DATA DIVISION.
000015 000 FILE SECTION.
000016 000 FD PRNTFL
000017 000 LABEL RECORD OMITTED
000018 000 DATA RECORD PRNTLINE.
000019 000 01 PRNTLINE.
000020 000 02 NUMB PIC 9(5).
000021 000 02 FILLER PIC X(2).
000022 000 02 MESSAGE PIC X(125).
000023 000 FD CARDFL
000024 000 LABEL RECORD OMITTED
000025 000 DATA RECORD CARD.
000026 000 01 CARD.
000027 000 02 TPSN PIC 9(5).
000028 000 02 FILLER PIC X(1).
000029 000 02 AUCON PIC 9(6).
000030 000 02 FILLER PIC X(66).
000031 000 FD UD5FL
000032 000 BLOCK CONTAINS 24 RECORDS
000033 000 RECORD CONTAINS 360 CHARACTERS
000034 000 LABEL RECORD STANDARD
000035 000 DATA RECORD IS UDS.
000036 000 01 UDS.
000037 000 03 RECDL PIC X.
000038 000 03 FICOD PIC X.
000039 000 03 COMPU PIC X.
000040 000 03 UICCC.
000041 000 04 UIC12 PIC X(2).
000042 000 04 UIC36 PIC X(4).
000043 000 03 EDATE PIC 9(6).
000044 000 03 ACTCO PIC A.
000045 000 03 AMSCD PIC X(11).
000046 000 03 ASGMT PIC X(2).
000047 000 03 AUTHR PIC X(15).
000048 000 03 BRNCH PIC A(2).
000049 000 03 CAR55 PIC X(2).
000050 000 03 CCNUM PIC X(6).
000051 000 03 CHGMR PIC X(2).
000052 000 03 CIVCN PIC X(4).
000053 000 03 DAMPL PIC X(5).
000054 000 03 DEPLO PIC X(3).
000055 000 03 UNPID PIC X(2).
```


AD-A048 554

ARMY CONCEPTS ANALYSIS AGENCY BETHESDA MD
COMPUTER ASSISTED MATCH PROGRAM (CAMP), (U)
AUG 76 G L MARTIN, E R MONTAGNE
CAA-D-76-5

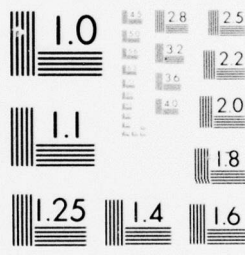
F/O 15/7

UNCLASSIFIED

NL

2 OF 4
AD-A048554





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

FMA/HRG PRE-PROCESSOR PROGRAM SORTUDS **UNCLASSIFIED**

```

WELT.L 32MARTIN.SORTUDS2FAS
ELT007 RL1870 08/16-07:40:09-(1,1)
000001 000 IDENTIFICATION DIVISION. SORTUDS2FAS
000002 001 PROGRAM-ID.
000003 000 ENVIRONMENT DIVISION.
000004 000 CONFIGURATION SECTION.
000005 000 SOURCE=COMPUTER. UNIVAC-1108.
000006 000 OBJECT=COMPUTER. UNIVAC-1108.
000007 000 INPUT-OUTPUT SECTION.
000008 000 FILE-CONTROL.
000009 000 SELECT PRNTFL ASSIGN TO PRINTER.
000010 000 SELECT CARDFL ASSIGN TO CARD-READER.
000011 000 SELECT UD5FL ASSIGN TO UNISERVO SRC-TAPE.
000012 000 SELECT FASFL ASSIGN TO UNISERVO UIC-TAPE.
000013 000 SELECT SRTFL ASSIGN TO MASS-STORAGE XA.
000014 000 DATA DIVISION.
000015 000 FILE SECTION.
000016 000 FD PRNTFL
000017 000 LABEL RECORD OMITTED
000018 000 DATA RECORD PRNTLINE.
000019 000 01 PRNTLINE.
000020 000 02 NUMB
000021 000 02 FILLER
000022 000 02 MESSAGE
000023 000 02 CARDFL
000024 000 LABEL RECORD OMITTED
000025 000 DATA RECORD CARD.
000026 000 01 CARD.
000027 000 02 TPSN
000028 000 02 FILLER
000029 000 02 AUDON
000030 000 02 FILLER
000031 000 UD5FL
000032 000 BLOCK CONTAINS 24 RECORDS
000033 000 RECORD CONTAINS 360 CHARACTERS
000034 000 LABEL RECORD STANDARD
000035 000 DATA RECORD 15 UDS.
000036 000 01 UDS.
000037 000 03 RECDL
000038 000 03 FICOD
000039 000 03 COMPO
000040 000 03 UICCC.
000041 000 04 UIC12
000042 000 04 UIC36
000043 000 03 EDATC
000044 000 03 ACTCO
000045 000 03 AMSCO
000046 000 03 X(11).
000047 000 03 X(12).
000048 000 03 X(14).
000049 000 03 X(16).
000050 000 03 X(18).
000051 000 03 X(20).
000052 000 03 X(22).
000053 000 03 X(24).
000054 000 03 X(26).
000055 000 03 X(28).
000056 000 03 X(30).
000057 000 03 X(32).
000058 000 03 X(34).
000059 000 03 X(36).
000060 000 03 X(38).
000061 000 03 X(40).
000062 000 03 X(42).
000063 000 03 X(44).
000064 000 03 X(46).
000065 000 03 X(48).
000066 000 03 X(50).
000067 000 03 X(52).
000068 000 03 X(54).
000069 000 03 X(56).
000070 000 03 X(58).
000071 000 03 X(60).
000072 000 03 X(62).
000073 000 03 X(64).
000074 000 03 X(66).
000075 000 03 X(68).
000076 000 03 X(70).
000077 000 03 X(72).
000078 000 03 X(74).
000079 000 03 X(76).
000080 000 03 X(78).
000081 000 03 X(80).
000082 000 03 X(82).
000083 000 03 X(84).
000084 000 03 X(86).
000085 000 03 X(88).
000086 000 03 X(90).
000087 000 03 X(92).
000088 000 03 X(94).
000089 000 03 X(96).
000090 000 03 X(98).
000091 000 03 X(100).
000092 000 03 X(102).
000093 000 03 X(104).
000094 000 03 X(106).
000095 000 03 X(108).
000096 000 03 X(110).
000097 000 03 X(112).
000098 000 03 X(114).
000099 000 03 X(116).
000100 000 03 X(118).
000101 000 03 X(120).
000102 000 03 X(122).
000103 000 03 X(124).
000104 000 03 X(126).
000105 000 03 X(128).
000106 000 03 X(130).
000107 000 03 X(132).
000108 000 03 X(134).
000109 000 03 X(136).
000110 000 03 X(138).
000111 000 03 X(140).
000112 000 03 X(142).
000113 000 03 X(144).
000114 000 03 X(146).
000115 000 03 X(148).
000116 000 03 X(150).
000117 000 03 X(152).
000118 000 03 X(154).
000119 000 03 X(156).
000120 000 03 X(158).
000121 000 03 X(160).
000122 000 03 X(162).
000123 000 03 X(164).
000124 000 03 X(166).
000125 000 03 X(168).
000126 000 03 X(170).
000127 000 03 X(172).
000128 000 03 X(174).
000129 000 03 X(176).
000130 000 03 X(178).
000131 000 03 X(180).
000132 000 03 X(182).
000133 000 03 X(184).
000134 000 03 X(186).
000135 000 03 X(188).
000136 000 03 X(190).
000137 000 03 X(192).
000138 000 03 X(194).
000139 000 03 X(196).
000140 000 03 X(198).
000141 000 03 X(200).
000142 000 03 X(202).
000143 000 03 X(204).
000144 000 03 X(206).
000145 000 03 X(208).
000146 000 03 X(210).
000147 000 03 X(212).
000148 000 03 X(214).
000149 000 03 X(216).
000150 000 03 X(218).
000151 000 03 X(220).
000152 000 03 X(222).
000153 000 03 X(224).
000154 000 03 X(226).
000155 000 03 X(228).
000156 000 03 X(230).
000157 000 03 X(232).
000158 000 03 X(234).
000159 000 03 X(236).
000160 000 03 X(238).
000161 000 03 X(240).
000162 000 03 X(242).
000163 000 03 X(244).
000164 000 03 X(246).
000165 000 03 X(248).
000166 000 03 X(250).
000167 000 03 X(252).
000168 000 03 X(254).
000169 000 03 X(256).
000170 000 03 X(258).
000171 000 03 X(260).
000172 000 03 X(262).
000173 000 03 X(264).
000174 000 03 X(266).
000175 000 03 X(268).
000176 000 03 X(270).
000177 000 03 X(272).
000178 000 03 X(274).
000179 000 03 X(276).
000180 000 03 X(278).
000181 000 03 X(280).
000182 000 03 X(282).
000183 000 03 X(284).
000184 000 03 X(286).
000185 000 03 X(288).
000186 000 03 X(290).
000187 000 03 X(292).
000188 000 03 X(294).
000189 000 03 X(296).
000190 000 03 X(298).
000191 000 03 X(300).
000192 000 03 X(302).
000193 000 03 X(304).
000194 000 03 X(306).
000195 000 03 X(308).
000196 000 03 X(310).
000197 000 03 X(312).
000198 000 03 X(314).
000199 000 03 X(316).
000200 000 03 X(318).
000201 000 03 X(320).
000202 000 03 X(322).
000203 000 03 X(324).
000204 000 03 X(326).
000205 000 03 X(328).
000206 000 03 X(330).
000207 000 03 X(332).
000208 000 03 X(334).
000209 000 03 X(336).
000210 000 03 X(338).
000211 000 03 X(340).
000212 000 03 X(342).
000213 000 03 X(344).
000214 000 03 X(346).
000215 000 03 X(348).
000216 000 03 X(350).
000217 000 03 X(352).
000218 000 03 X(354).
000219 000 03 X(356).
000220 000 03 X(358).
000221 000 03 X(360).
000222 000 03 X(362).
000223 000 03 X(364).
000224 000 03 X(366).
000225 000 03 X(368).
000226 000 03 X(370).
000227 000 03 X(372).
000228 000 03 X(374).
000229 000 03 X(376).
000230 000 03 X(378).
000231 000 03 X(380).
000232 000 03 X(382).
000233 000 03 X(384).
000234 000 03 X(386).
000235 000 03 X(388).
000236 000 03 X(390).
000237 000 03 X(392).
000238 000 03 X(394).
000239 000 03 X(396).
000240 000 03 X(398).
000241 000 03 X(400).
000242 000 03 X(402).
000243 000 03 X(404).
000244 000 03 X(406).
000245 000 03 X(408).
000246 000 03 X(410).
000247 000 03 X(412).
000248 000 03 X(414).
000249 000 03 X(416).
000250 000 03 X(418).
000251 000 03 X(420).
000252 000 03 X(422).
000253 000 03 X(424).
000254 000 03 X(426).
000255 000 03 X(428).
000256 000 03 X(430).
000257 000 03 X(432).
000258 000 03 X(434).
000259 000 03 X(436).
000260 000 03 X(438).
000261 000 03 X(440).
000262 000 03 X(442).
000263 000 03 X(444).
000264 000 03 X(446).
000265 000 03 X(448).
000266 000 03 X(450).
000267 000 03 X(452).
000268 000 03 X(454).
000269 000 03 X(456).
000270 000 03 X(458).
000271 000 03 X(460).
000272 000 03 X(462).
000273 000 03 X(464).
000274 000 03 X(466).
000275 000 03 X(468).
000276 000 03 X(470).
000277 000 03 X(472).
000278 000 03 X(474).
000279 000 03 X(476).
000280 000 03 X(478).
000281 000 03 X(480).
000282 000 03 X(482).
000283 000 03 X(484).
000284 000 03 X(486).
000285 000 03 X(488).
000286 000 03 X(490).
000287 000 03 X(492).
000288 000 03 X(494).
000289 000 03 X(496).
000290 000 03 X(498).
000291 000 03 X(500).
000292 000 03 X(502).
000293 000 03 X(504).
000294 000 03 X(506).
000295 000 03 X(508).
000296 000 03 X(510).
000297 000 03 X(512).
000298 000 03 X(514).
000299 000 03 X(516).
000300 000 03 X(518).
000301 000 03 X(520).
000302 000 03 X(522).
000303 000 03 X(524).
000304 000 03 X(526).
000305 000 03 X(528).
000306 000 03 X(530).
000307 000 03 X(532).
000308 000 03 X(534).
000309 000 03 X(536).
000310 000 03 X(538).
000311 000 03 X(540).
000312 000 03 X(542).
000313 000 03 X(544).
000314 000 03 X(546).
000315 000 03 X(548).
000316 000 03 X(550).
000317 000 03 X(552).
000318 000 03 X(554).
000319 000 03 X(556).
000320 000 03 X(558).
000321 000 03 X(560).
000322 000 03 X(562).
000323 000 03 X(564).
000324 000 03 X(566).
000325 000 03 X(568).
000326 000 03 X(570).
000327 000 03 X(572).
000328 000 03 X(574).
000329 000 03 X(576).
000330 000 03 X(578).
000331 000 03 X(580).
000332 000 03 X(582).
000333 000 03 X(584).
000334 000 03 X(586).
000335 000 03 X(588).
000336 000 03 X(590).
000337 000 03 X(592).
000338 000 03 X(594).
000339 000 03 X(596).
000340 000 03 X(598).
000341 000 03 X(600).
000342 000 03 X(602).
000343 000 03 X(604).
000344 000 03 X(606).
000345 000 03 X(608).
000346 000 03 X(610).
000347 000 03 X(612).
000348 000 03 X(614).
000349 000 03 X(616).
000350 000 03 X(618).
000351 000 03 X(620).
000352 000 03 X(622).
000353 000 03 X(624).
000354 000 03 X(626).
000355 000 03 X(628).
000356 000 03 X(630).
000357 000 03 X(632).
000358 000 03 X(634).
000359 000 03 X(636).
000360 000 03 X(638).
000361 000 03 X(640).
000362 000 03 X(642).
000363 000 03 X(644).
000364 000 03 X(646).
000365 000 03 X(648).
000366 000 03 X(650).
000367 000 03 X(652).
000368 000 03 X(654).
000369 000 03 X(656).
000370 000 03 X(658).
000371 000 03 X(660).
000372 000 03 X(662).
000373 000 03 X(664).
000374 000 03 X(666).
000375 000 03 X(668).
000376 000 03 X(670).
000377 000 03 X(672).
000378 000 03 X(674).
000379 000 03 X(676).
000380 000 03 X(678).
000381 000 03 X(680).
000382 000 03 X(682).
000383 000 03 X(684).
000384 000 03 X(686).
000385 000 03 X(688).
000386 000 03 X(690).
000387 000 03 X(692).
000388 000 03 X(694).
000389 000 03 X(696).
000390 000 03 X(698).
000391 000 03 X(700).
000392 000 03 X(702).
000393 000 03 X(704).
000394 000 03 X(706).
000395 000 03 X(708).
000396 000 03 X(710).
000397 000 03 X(712).
000398 000 03 X(714).
000399 000 03 X(716).
000400 000 03 X(718).
000401 000 03 X(720).
000402 000 03 X(722).
000403 000 03 X(724).
000404 000 03 X(726).
000405 000 03 X(728).
000406 000 03 X(730).
000407 000 03 X(732).
000408 000 03 X(734).
000409 000 03 X(736).
000410 000 03 X(738).
000411 000 03 X(740).
000412 000 03 X(742).
000413 000 03 X(744).
000414 000 03 X(746).
000415 000 03 X(748).
000416 000 03 X(750).
000417 000 03 X(752).
000418 000 03 X(754).
000419 000 03 X(756).
000420 000 03 X(758).
000421 000 03 X(760).
000422 000 03 X(762).
000423 000 03 X(764).
000424 000 03 X(766).
000425 000 03 X(768).
000426 000 03 X(770).
000427 000 03 X(772).
000428 000 03 X(774).
000429 000 03 X(776).
000430 000 03 X(778).
000431 000 03 X(780).
000432 000 03 X(782).
000433 000 03 X(784).
000434 000 03 X(786).
000435 000 03 X(788).
000436 000 03 X(790).
000437 000 03 X(792).
000438 000 03 X(794).
000439 000 03 X(796).
000440 000 03 X(798).
000441 000 03 X(800).
000442 000 03 X(802).
000443 000 03 X(804).
000444 000 03 X(806).
000445 000 03 X(808).
000446 000 03 X(810).
000447 000 03 X(812).
000448 000 03 X(814).
000449 000 03 X(816).
000450 000 03 X(818).
000451 000 03 X(820).
000452 000 03 X(822).
000453 000 03 X(824).
000454 000 03 X(826).
000455 000 03 X(828).
000456 000 03 X(830).
000457 000 03 X(832).
000458 000 03 X(834).
000459 000 03 X(836).
000460 000 03 X(838).
000461 000 03 X(840).
000462 000 03 X(842).
000463 000 03 X(844).
000464 000 03 X(846).
000465 000 03 X(848).
000466 000 03 X(850).
000467 000 03 X(852).
000468 000 03 X(854).
000469 000 03 X(856).
000470 000 03 X(858).
000471 000 03 X(860).
000472 000 03 X(862).
000473 000 03 X(864).
000474 000 03 X(866).
000475 000 03 X(868).
000476 000 03 X(870).
000477 000 03 X(872).
000478 000 03 X(874).
000479 000 03 X(876).
000480 000 03 X(878).
000481 000 03 X(880).
000482 000 03 X(882).
000483 000 03 X(884).
000484 000 03 X(886).
000485 000 03 X(888).
000486 000 03 X(890).
000487 000 03 X(892).
000488 000 03 X(894).
000489 000 03 X(896).
000490 000 03 X(898).
000491 000 03 X(900).
000492 000 03 X(902).
000493 000 03 X(904).
000494 000 03 X(906).
000495 000 03 X(908).
000496 000 03 X(910).
000497 000 03 X(912).
000498 000 03 X(914).
000499 000 03 X(916).
000500 000 03 X(918).
000501 000 03 X(920).
000502 000 03 X(922).
000503 000 03 X(924).
000504 000 03 X(926).
000505 000 03 X(928).
000506 000 03 X(930).
000507 000 03 X(932).
000508 000 03 X(934).
000509 000 03 X(936).
000510 000 03 X(938).
000511 000 03 X(940).
000512 000 03 X(942).
000513 000 03 X(944).
000514 000 03 X(946).
000515 000 03 X(948).
000516 000 03 X(950).
000517 000 03 X(952).
000518 000 03 X(954).
000519 000 03 X(956).
000520 000 03 X(958).
000521 000 03 X(960).
000522 000 03 X(962).
000523 000 03 X(964).
000524 000 03 X(966).
000525 000 03 X(968).
000526 000 03 X(970).
000527 000 03 X(972).
000528 000 03 X(974).
000529 000 03 X(976).
000530 000 03 X(978).
000531 000 03 X(980).
000532 000 03 X(982).
000533 000 03 X(984).
000534 000 03 X(986).
000535 000 03 X(988).
000536 000 03 X(990).
000537 000 03 X(992).
000538 000 03 X(994).
000539 000 03 X(996).
000540 000 03 X(998).
000541 000 03 X(1000).
000542 000 03 X(1002).
000543 000 03 X(1004).
000544 000 03 X(1006).
000545 000 03 X(1008).
000546 000 03 X(1010).
000547 000 03 X(1012).
000548 000 03 X(1014).
000549 000 03 X(1016).
000550 000 03 X(1018).
000551 000 03 X(1020).
000552 000 03 X(1022).
000553 000 03 X(1024).
000554 000 03 X(1026).
000555 000 03 X(1028).
000556 000 03 X(1030).
000557 000 03 X(1032).
000558 000 03 X(1034).
000559 000 03 X(1036).
000560 000 03 X(1038).
000561 000 03 X(1040).
000562 000 03 X(1042).
000563 000 03 X(1044).
000564 000 03 X(1046).
000565 000 03 X(1048).
000566 000 03 X(1050).
000567 000 03 X(1052).
000568 000 03 X(1054).
000569 000 03 X(1056).
000570 000 03 X(1058).
000571 000 03 X(1060).
000572 000 03 X(1062).
000573 000 03 X(1064).
000574 000 03 X(1066).
000575 000 03 X(1068).
000576 000 03 X(1070).
000577 000 03 X(1072).
000578 000 03 X(1074).
000579 000 03 X(1076).
000580 000 03 X(1078).
000581 000 03 X(1080).
000582 000 03 X(1082).
000583 000 03 X(1084).
000584 000 03 X(1086).
000585 000 03 X(1088).
000586 000 03 X(1090).
000587 000 03 X(1092).
000588 000 03 X(1094).
000589 000 03 X(1096).
000590 000 03 X(1098).
000591 000 03 X(1100).
000592 000 03 X(1102).
000593 000 03 X(1104).
000594 000 03 X(1106).
000595 000 03 X(1108).
000596 000 03 X(1110).
000597 000 03 X(1112).
000598 000 03 X(1114).
000599 000 03 X(1116).
000600 000 03 X(1118).
000601 000 03 X(1120).
000602 000 03 X(1122).
000603 000 03 X(1124).
000604 000 03 X(1126).
000605 000 03 X(1128).
000606 000 03 X(1130).
000607 000 03 X(1132).
000608 000 03 X(1134).
000609 000 03 X(1136).
000610 000 03 X(1138).
000611 000 03 X(1140).
000612 000 03 X(1142).
000613 000 03 X(1144).
000614 000 03 X(1146).

```

000056	000	03 MBPRD	PIC X(3).
000057	000	03 DSCMP	PIC X(2).
000058	000	03 ELSEW	PIC X(2).
000059	000	03 EQCON	PIC X.
000060	000	03 ESCON	PIC X.
000061	000	03 ELAGG	PIC X.
000062	000	03 FNCAT	PIC X.
000063	000	03 FURCO	PIC X.
000064	000	03 FPLAN.	
000065	000	04 FPLAN1	PIC X.
000066	000	04 FPLAN2	PIC X.
000067	000	04 FPLAN3	PIC X.
000068	000	03 JCSTY	PIC X(3).
000069	000	03 LOCCO	PIC X(3).
000070	000	03 MACTO	PIC X(3).
000071	000	03 MRCMD	PIC X(2).
000072	000	03 MBLOC	PIC X(3).
000073	000	03 FILLR	PIC X(2).
000074	000	03 MBSTA	PIC X(9).
000075	000	03 MILCN	PIC X(2).
000076	000	03 NOTID	PIC X.
000077	000	03 NTREF	PIC X(2).
000078	000	03 OESTS	PIC X.
000079	000	03 OPAGY	PIC 9(3).
000080	000	03 ADCON.	
000081	000	04 ADC01	PIC 9.
000082	000	04 ADC02	PIC 99.
000083	000	04 ADC03	PIC 999.
000084	000	03 OPDAT	PIC X.
000085	000	03 CATCO	PIC X.
000086	000	03 PHASE	PIC X.
000087	000	03 PRCON	PIC X.
000088	000	03 REPCO	PIC X.
000089	000	03 ROBCO.	
000090	000	04 ROBC01	PIC X.
000091	000	04 ROBC02	PIC X.
000092	000	04 ROBC03	PIC X.
000093	000	03 SPLIT	PIC A.
000094	000	03 SRCO	PIC X(13).
000095	000	03 STACO	PIC X(5).
000096	000	03 STATS	PIC X(2).
000097	000	03 STNM	PIC X(9).
000098	000	03 TDATE	PIC 9(5).
000099	000	03 TDNUM	PIC X(11).
000100	000	03 TMCCC	PIC A.
000101	000	03 TPSNA	PIC 9(5).
000102	000	03 TRCON	PIC X.
000103	000	03 TYPCO	PIC 9.
000104	000	03 UNCAP	PIC X.
000105	000	03 UNCON	PIC X.
000106	000	03 UNCLC	PIC X(5).
000107	000	03 UNMBR	PIC X(4).
000108	000	03 PECOD	PIC X(6).
000109	000	03 ULCCC	PIC A(3).
000110	000	03 UNTDS	PIC X(18).
000111	000	03 VCHNR	PIC X(3).
000112	000	03 AUSTR	PIC X(25).

000113	000	03	OPSTR	PIC X(25).
000114	000	03	STSTR	PIC X(25).
000115	000	03	DPMT	PIC X(4).
000116	000	03	ENDRC	PIC X.
000117	000	03	FILLER	PIC X(52).
000118	000	03	FASFL	
000119	000	03	RECORD CONTAINS 7368 CHARACTERS	
000120	000	03	LABEL RECORD OMITTED	
000121	000	03	RECORDING MODE IS 1	
000122	000	03	DATA RECORD IS FAS-BLK.	
000123	000	03	FAS-BLK.	
000124	000	03	FAS-REC OCCURS 24 TIMES.	
000125	000	03	FAS-REC OCCURS 24 TIMES.	
000126	000	03	FAS-REC OCCURS 24 TIMES.	
000127	000	03	FAS-REC OCCURS 24 TIMES.	
000128	000	03	FAS-REC OCCURS 24 TIMES.	
000129	000	03	FAS-REC OCCURS 24 TIMES.	
000130	000	03	FAS-REC OCCURS 24 TIMES.	
000131	000	03	FAS-REC OCCURS 24 TIMES.	
000132	000	03	FAS-REC OCCURS 24 TIMES.	
000133	000	03	FAS-REC OCCURS 24 TIMES.	
000134	000	03	FAS-REC OCCURS 24 TIMES.	
000135	000	03	FAS-REC OCCURS 24 TIMES.	
000136	000	03	FAS-REC OCCURS 24 TIMES.	
000137	000	03	FAS-REC OCCURS 24 TIMES.	
000138	000	03	FAS-REC OCCURS 24 TIMES.	
000139	000	03	FAS-REC OCCURS 24 TIMES.	
000140	000	03	FAS-REC OCCURS 24 TIMES.	
000141	000	03	FAS-REC OCCURS 24 TIMES.	
000142	000	03	FAS-REC OCCURS 24 TIMES.	
000143	000	03	FAS-REC OCCURS 24 TIMES.	
000144	000	03	FAS-REC OCCURS 24 TIMES.	
000145	000	03	FAS-REC OCCURS 24 TIMES.	
000146	000	03	FAS-REC OCCURS 24 TIMES.	
000147	000	03	FAS-REC OCCURS 24 TIMES.	
000148	000	03	FAS-REC OCCURS 24 TIMES.	
000149	000	03	FAS-REC OCCURS 24 TIMES.	
000150	000	03	FAS-REC OCCURS 24 TIMES.	
000151	000	03	FAS-REC OCCURS 24 TIMES.	
000152	000	03	FAS-REC OCCURS 24 TIMES.	
000153	000	03	FAS-REC OCCURS 24 TIMES.	
000154	000	03	FAS-REC OCCURS 24 TIMES.	
000155	000	03	FAS-REC OCCURS 24 TIMES.	
000156	000	03	FAS-REC OCCURS 24 TIMES.	
000157	000	03	FAS-REC OCCURS 24 TIMES.	
000158	000	03	FAS-REC OCCURS 24 TIMES.	
000159	000	03	FAS-REC OCCURS 24 TIMES.	
000160	000	03	FAS-REC OCCURS 24 TIMES.	
000161	000	03	FAS-REC OCCURS 24 TIMES.	
000162	000	03	FAS-REC OCCURS 24 TIMES.	
000163	000	03	FAS-REC OCCURS 24 TIMES.	
000164	000	03	FAS-REC OCCURS 24 TIMES.	
000165	000	03	FAS-REC OCCURS 24 TIMES.	
000166	000	03	FAS-REC OCCURS 24 TIMES.	
000167	000	03	FAS-REC OCCURS 24 TIMES.	
000168	000	03	FAS-REC OCCURS 24 TIMES.	
000169	000	03	FAS-REC OCCURS 24 TIMES.	

000170	000	03	MBLMD	PIC X(2).
000171	000	03	MBLOC	PIC X(3).
000172	000	03	FILLR	PIC X(2).
000173	000	03	MBSTA	PIC X(9).
000174	000	03	MILCN	PIC X(2).
000175	000	03	NOTID	PIC X.
000176	000	03	NTRF	PIC X(2).
000177	000	03	OESTS	PIC X.
000178	000	03	OPAGY	PIC 9(1).
000179	000	03	ADCON.	
000180	000	04	ADC01	PIC 9.
000181	000	04	ADC02	PIC 99.
000182	000	04	ADC03	PIC 999.
000183	000	03	OPDAT	PIC X.
000184	000	03	CATCO	PIC X.
000185	000	03	PHASE	PIC X.
000186	000	03	PRCON	PIC X.
000187	000	03	REPCO	PIC X.
000188	000	03	ROBCO.	
000189	000	04	ROBC01	PIC X.
000190	000	04	ROBC02	PIC X.
000191	000	04	ROBC03	PIC X.
000192	000	03	SPLIT	PIC A.
000193	000	03	SRC0	PIC X(13).
000194	000	03	STACO	PIC X(5).
000195	000	03	STATS	PIC X(2).
000196	000	03	STNMH	PIC X(9).
000197	000	03	TDATE	PIC 9(5).
000198	000	03	TNUM	PIC X(11).
000199	000	03	TMCCC	PIC A.
000200	000	03	TPSNA	PIC 9(5).
000201	000	03	TRCON	PIC X.
000202	000	03	TYPCO	PIC 9.
000203	000	03	UNCAP	PIC X.
000204	000	03	UNCON	PIC X.
000205	000	03	UNCLC	PIC X(5).
000206	000	03	UNMBR	PIC X(4).
000207	000	03	PECOD	PIC X(6).
000208	000	03	ULCCC	PIC A(3).
000209	000	03	UNIDS	PIC X(18).
000210	000	03	VMNR	PIC X(3).
000211	000	03	AUSTR	PIC X(25).
000212	000	03	OPSTR	PIC X(25).
000213	000	03	STSTR	PIC X(25).
000214	000	03	DPINT	PIC X(4).
000215	000	03	ENDRC	PIC X.
000216	000	WORKING-STORAGE SECTION.		
000217	000	77	BRC	PIC 9(5).
000218	000	01	WORK-SRC.	
000219	000	02	WSRC1-5.	
000220	000	03	WSRC1-3	PIC X(3).
000221	000	03	WSRC4-5	PIC X(2).
000222	000	02	WSRC612.	
000223	000	03	WSRC6-9	PIC X(4).
000224	000	03	WSRCALO	PIC X(1).
000225	000	03	WSRC11-12	PIC X(2).
000226	000	01	FAS.	

000227	000	03	RECEL	PIC X*
000228	000	03	FICOD	PIC X*
000229	000	03	COMPO	PIC X*
000230	000	03	UICCC	
000231	000	04	UIC12	PIC X(2)*
000232	000	04	UIC36	PIC X(4)*
000233	000	03	EDATE	PIC 9(6)*
000234	000	03	ACTCO	PIC A*
000235	000	03	AMSCO	PIC X(10)*
000236	000	03	ASSMT	PIC X(2)*
000237	000	03	AUTHR	
000238	000	04	UIN	PIC X(3)*
000239	000	04	FLG	PIC X(4)*
000240	000	04	MATCH	PIC X*
000241	000	04	TP	PIC 9(2)*
000242	000	04	RDD	PIC 9(3)*
000243	000	04	LRE	PIC X*
000244	000	04	THR	PIC 9*
000245	000	03	BRNCH	PIC A(2)*
000246	000	03	CANSS	PIC X(2)*
000247	000	03	CCNUM	PIC X(6)*
000248	000	03	CHGHR	PIC X(2)*
000249	000	03	CIVCN	PIC X(2)*
000250	000	03	DAMPL	PIC X(5)*
000251	000	03	DEPLO	PIC X(3)*
000252	000	03	UNPID	PIC X(2)*
000253	000	03	MBPRD	PIC X(3)*
000254	000	03	DSOMP	PIC X(2)*
000255	000	03	ELSEQ	PIC X(2)*
000256	000	03	EGCON	PIC X*
000257	000	03	ESCON	PIC X*
000258	000	03	ELAGG	PIC X*
000259	000	03	FNGAT	PIC X*
000260	000	03	FORCO	PIC X*
000261	000	03	FPLAN*	
000262	000	04	FPLAN1	PIC X*
000263	000	04	FPLAN2	PIC X*
000264	000	04	FPLAN3	PIC X*
000265	000	03	JCSTY	PIC X(5)*
000266	000	03	LOCCO	PIC X(3)*
000267	000	03	MACTO	PIC X(3)*
000268	000	03	MBCHD	PIC X(2)*
000269	000	03	MBLOC	PIC X(3)*
000270	000	03	FILLR	PIC X(2)*
000271	000	03	MBSTA	PIC X(9)*
000272	000	03	MILCN	PIC X(2)*
000273	000	03	NOTID	PIC X*
000274	000	03	NMEF	PIC X(2)*
000275	000	03	OESTS	PIC X*
000276	000	03	OPAGY	PIC 9(3)*
000277	000	03	ADCON*	
000278	000	04	ADC01	PIC 9*
000279	000	04	ADC02	PIC 99*
000280	000	04	ADC03	PIC 999*
000281	000	03	OPDAT	PIC X*
000282	000	03	CATCO	PIC X*
000283	000	03	PHASE	PIC X*

000284	000	03	PRCON	PIC X.
000285	000	03	REPCO	PIC X.
000286	000	03	ROBCO.	
000287	000	04	ROBCO1	PIC X.
000288	000	04	ROBCO2	PIC X.
000289	000	04	ROBCO3	PIC X.
000290	000	03	SPLIT	PIC A.
000291	000	03	SKCTO	PIC X(13).
000292	000	03	STACO	PIC X(5).
000293	000	03	STATS	PIC X(12).
000294	000	03	STNM	PIC X(9).
000295	000	03	TDATE	PIC 9(5).
000296	000	03	TNUM	PIC X(11).
000297	000	03	TMCCC	PIC A.
000298	000	03	TPSNA	PIC 9(5).
000299	000	03	TRCON	PIC X.
000300	000	03	TYPCO	PIC 9.
000301	000	03	UNCAP	PIC X.
000302	000	03	UNCON	PIC X.
000303	000	03	UNCLC	PIC X(5).
000304	000	03	UNMBR	PIC X(4).
000305	000	03	PECCD	PIC X(6).
000306	000	03	ULCCC	PIC A(3).
000307	000	03	UNDS	PIC X(18).
000308	000	03	VCHNR	PIC X(3).
000309	000	03	AUSTR	PIC X(25).
000310	000	03	OPSTR	PIC X(25).
000311	000	03	STSTR	PIC X(25).
000312	000	03	OPMNT	PIC X(4).
000313	000	03	ENDRC	PIC X.
000314	000	01	PRNTWRK.	
000315	000	03	UICCC.	
000316	000	04	UIC12	PIC X(2).
000317	000	04	UIC36	PIC X(4).
000318	000	03	FILLER	PIC XX VALUE SPACES.
000319	000	03	TPSNA	PIC 9(5).
000320	000	03	FILLER	PIC XX VALUE SPACES.
000321	000	03	COMPO	PIC X.
000322	000	03	FILLER	PIC XX VALUE SPACES.
000323	000	03	FPLAN.	
000324	000	04	FPLAN1	PIC X.
000325	000	04	FPLAN2	PIC X.
000326	000	04	FPLAN3	PIC X.
000327	000	03	FILLER	PIC XX VALUE SPACES.
000328	000	03	ROBCO.	
000329	000	04	ROBCO1	PIC X.
000330	000	04	ROBCO2	PIC X.
000331	000	04	ROBCO3	PIC X.
000332	000	03	FILLER	PIC XX VALUE SPACES.
000333	000	03	SKCTO	PIC X(12).
000334	000	03	FILLER	PIC XX VALUE SPACES.
000335	000	03	JCSTY	PIC X(5).
000336	000	03	FILLER	PIC XX VALUE SPACES.
000337	000	03	ULCCC	PIC X(3).
000338	000	03	FILLER	PIC XX VALUE SPACES.
000339	000	03	CARSS	PIC X(2).
000340	000	03	FILLER	PIC XX VALUE SPACES.


```
000341 000 03 UNBR PIC X(4).
000342 000 03 FILLER PIC XX VALUE SPACES.
000343 000 03 BRNCH PIC X(2).
000344 000 03 FILLER PIC XX VALUE SPACES.
000345 000 03 UNDS PIC X(15).
000346 000 03 FILLER PIC X VALUE SPACE.
000347 000 03 MBSTA PIC X(9).
000348 000 03 FILLER PIC X VALUE SPACE.
000349 000 03 MBLOC PIC X(3).
000350 000 03 FILLER PIC X VALUE SPACE.
000351 000 03 STNM PIC X(9).
000352 000 03 FILLER PIC X VALUE SPACE.
000353 000 03 LOCO PIC X(3).
000354 000 03 FILLER PIC X VALUE SPACE.
000355 000 03 DSCMP PIC X(2).
000356 000 03 FILLER PIC XX VALUE SPACES.
000357 000 03 ADCON.
000358 000 04 ADC01 PIC 9.
000359 000 04 FILLER PIC X VALUE SPACES.
000360 000 04 ADC02 PIC 99.
000361 000 04 FILLER PIC X VALUE SPACES.
000362 000 04 ADC03 PIC 999.
000363 000 PROCEDURE DIVISION.
000364 000 OPENERS.
000365 000 OPEN INPUT UDSFL CARDFL OUTPUT FASFL PRNTFL.
000366 000 READ CARDFL AT END MOVE 99999 TO TPSN IN CARD.
000367 000 MOVE ZERO TO BRC.
000368 000 SORT=IT.
000369 001 SORT SRTFL ON ASCENDING KEY COMPO UICCC
000370 000 INPUT PROCEDURE IS SORT-IN THRU UDS-EOF
000371 000 OUTPUT PROCEDURE IS SORT-OUT.
000372 000 GO TO CLOSEOUT.
000373 000 SORT-IN.
000374 000 READ UDSFL AT END GO TO UDS-EOF.
000375 000 MOVE CORRESPONDING UDS TO FAS.
000376 000 RELEASE SRT FROM FAS.
000377 000 GO TO SORT-IN.
000378 000 UDS-EOF.
000379 000 EXIT.
000380 000 SORT-OUT.
000381 000 RETURN SRTFL INTO FAS AT END GO TO LAST-BLK.
000382 000 IF TPSN IN CARD EQUALS TPSNA IN FAS
000383 000 MOVE ADCON IN CARD TO ADCON IN FAS.
000384 000 MOVE CORRESPONDING FAS TO PRNT=OKK.
000385 000 WRITE PRNTLINE FROM PRNTWORK.
000386 000 ENTER FORTRAN FCOPY SUBROUTINE REFERENCING FAS.
000387 000 ADD 1 BRC.
000388 000 MOVE FAS TO FAS-REC (BRC).
000389 000 IF BRC EQUALS 24 WRITE FAS-BLK MOVE ZERO TO BRC.
000390 000 GO TO SORT-OUT.
000391 000 LAST-BLK.
000392 000 IF BRC EQUALS ZERO GO TO EOF-FAS.
000393 000 ADD 1 TO BRC.
000394 000 MOVE ALL '9' TO FAS=DATA (BRC).
000395 000 IF BRC LESS THAN 24 GO TO LAST-BLK.
000396 000 WRITE FAS-BLK.
000397 000 EOF-FAS.
```

```
000398 000 ENTER FORTRAN ECOPY SUBROUTINE REFERENCING FAS.  
000399 000 CLOSEOUT.  
000400 000 CLOSE UDSEFL FASEL PRNTFL CARDEL.  
000401 000 STOP RUN.
```

END ELT.

```
000402 000 32MARTIN.FCOPY  
ELT007 RL1870 08/16-07:40:13-(0,1)  
000001 000 SUBROUTINE FCOPY(FAS)  
000002 000 DIMENSION FAS(52)  
000003 000 DATA IL/0/  
000004 000 C  
000005 000 WRITE(2)FAS  
000006 000 IL=IL*1  
000007 000 IF(IL/500*500*EQ*IL)PRINT 1,IL  
000008 000 1 FORMAT(110)  
000009 000 RETURN  
000010 000 C  
000011 000 C  
000012 000 C  
000013 000 C  
000014 000 PUT EOF ON BINARY FILE  
000015 000 ENTRY FCOPY(FAS)  
000016 000 PRINT 99,IL  
000017 000 99 FORMAT()  
000018 000 ENDFILE 2  
000019 000 RETURN  
000020 000 C  
000021 000 C  
000022 000 C  
000023 000 READ BINARY FILE  
000024 000 ENTRY FREAD(FAS)  
000025 000 READ(2,END=999) FAS  
000026 000 RETURN  
000027 000 FAS(1)=6H9999999  
000028 000 FAS(2)=6H9999999  
000029 000 REMIND 2  
000030 000 RETURN  
000031 000 END
```

END ELT.

000402 FMA PROGRAM FSORT **UNCLASSIFIED** .L,0

```
SEL, L FSFS, CKALT
ELT007 RL1870 08/16-07:40:15-13, )
000001 003 SUBROUTINE CKALT (ADCO1, SIZE, ALTTAB, ALTFLG)
000002 000 IMPLICIT INTEGER (A-Z)
000003 000 INTEGER ALTTAB(100)
000004 000 LOGICAL ALTFLG
000005 002 C WRITE (6,1) ADCO1, SIZE, (ALTTAB(K), K=1, SIZE)
000006 002 1 FORMAT (' SUB CKALT ADCO1=', I4, ', SIZE=', I4, ' IX, 5A6)
000007 000 C
000008 000 ALTFLG = .FALSE.
000009 000 I = 1
000010 000 C
000011 000 C DO WHILE (NOT END OF TABLE ALTTAB)
000012 000 C
000013 000 1000 IF (I .GT. SIZE) RETURN
000014 000 C
000015 000 IF (ADCO1 .NE. ALTTAB(I)) GOTO 1100
000016 000 C
000017 000 ALTFLG = .TRUE.
000018 000 RETURN
000019 000 C ELSE
000020 000 1100 I = I + 1
000021 000 C END-IF
000022 000 GOTO 1000
000023 000 C END-DO
000024 000 END

END ELT.
```

```
SEL, L FSFS, CKHPU
ELT007 RL1870 08/16-07:40:16-(1, )
000001 000 C C K H P U - R U T I N E
000002 000 C
000003 000 C
000004 000 C
000005 000 C
000006 000 C THIS ROUTINE CHECKS THE CURRENT FASREC VIA THE INFORMATION CONTAINED IN
000007 000 C HPINFO AGAINST THE INFORMATION CONTAINED IN THE HAND-PLAYED-UNIT TABLE +PUTA+
000008 000 C TO DETERMINE IF THE FASREC IS TO BE HAND-PLAYED.
000009 000 C A GIVEN FASREC IS HAND-PLAYED IF ITS UIC (HPINFO(1)) EQUALS A HAND-PLAYED
000010 000 C UIC FROM THE HAND-PLAYED TABLE (HPUTAB(1)), WHERE 0 < I < LIMIT +1.
000011 000 C
000012 000 C GIVEN THAT THE FASREC IS A HAND-PLAYED-UNIT (UIC'S MATCH) THEN ROBCO, ADCO1
000013 000 C ADCO2, AND ADCO3 ARE SET EQUAL TO THE CORRESPONDING FIELD FROM HPUTAB IF THE
000014 000 C FIELD IN HPUTAB IS NON-BLANK. ALSO IF ADCO3 FIELD IS CHANGED THEN RDDFLG IS
000015 000 C SET TO .TRUE. TO INDICATE THAT THE RDD IS BEING HAND-PLAYED FOR THIS UNIT
000016 000 C
000017 000 C SUBROUTINE CKHPU (UIC, ROBCO, ADCO1, ADCO2, ADCO3, LIMIT,
000018 000 C HPUTAB, RDDFLG)
000019 000 C IMPLICIT INTEGER (A-Z)
000020 000 C INTEGER HPUTAB(100,6)
000021 000 C DATA BLANK/6H /
000022 000 C LOGICAL RDDFLG
000023 000 C
000024 000 C WRITE(6,1) UIC, ROBCO, ADCO1, ADCO2, ADCO3, LIMIT
```

RENE, R. PLOURDE
USACAA/MRD JUN 1976

UNCLASSIFIED

```

000025 000      1  FORMAT(' SUB CPU  UIC=',A6,'', ROBCO=',A3,'', ADCO1=',A1,
000026 000      * ', ADCO2=',A2,'', ADCO3=',A3,'', LIMIT=',A4)
000027 000      C
000028 000      C  HPUTAB(1,1) = UIC,  HPUTAB(1,2) = ROBCO,  HPUTAB(1,3) = ADCO1,
000029 000      C  HPUTAB(1,4) = ADCO2,  HPUTAB(1,5) = ADCO3,  HPUTAB(1,6) = FLAG
000030 000      C
000031 000      C  FIRST SET RUDFLG TO FALSE THEN CHECK TO SEE IF HPUTAB IS EMPTY
000032 000      C
000033 000      RUDFLG = .FALSE.
000034 000      IF (LIMIT .EQ. 0)  RETURN
000035 000      I = 1
000036 000      C
000037 000      C  DOWHILE (I < LIMIT)
000038 000      C
000039 000      1000 IF (I .GT. LIMIT)  RETURN
000040 000      C
000041 000      C  COMPARE UIC'S (FASREC UIC (HPINFO(1)) AND HAND-PLAYED (HPUTAB(1,1))
000042 000      C
000043 000      C  VICFLG = KUIC (UIC,HPUTAB(1,1))
000044 000      IF (UIC .EQ. HPUTAB(1,1))  GOTO 1100
000045 000      C  THEN
000046 000      C  UIC'S DID NOT MATCH THEREFORE INCREMENT POINTER AND CHECK NEXT ONE
000047 000      C
000048 000      I = I + 1
000049 000      GOTO 1900
000050 000      C  ELSE
000051 000      C
000052 000      C  UIC'S MATCH THEREFORE CHECK ROBCO AND ADCO1 FIELDS FROM HPUTAB
000053 000      C  AND CHANGE HPINFO FIELDS WHEN HPUTAB FIELDS ARE NON-BLANK
000054 000      C  CHECK HAND-PLAYED ROBCO FIELD FOR NON-BLANK
000055 000      C  IF (HPUTAB(1,2) .EQ. BLANK)  GOTO 1200
000056 000      C  THEN
000057 000      C  CHANGE FASREC ROBCO TO HAND-PLAYED ROBCO
000058 000      C  ROBCO = HPUTAB(1,2)
000059 000      C  ELSE
000060 000      C  HAND-PLAYED ROBCO IS BLANK THUS FASREC ROBCO IS NOT CHG'D
000061 000      C  END-IF
000062 000      C  CHECK HAND-PLAYED ADCO1 FIELD FOR NON-BLANK
000063 000      C  IF (HPUTAB(1,3) .EQ. BLANK)  GOTO 1300
000064 000      C  THEN
000065 000      C  CHANGE FASREC ADCO1 TO HAND-PLAYED ADCO1
000066 000      C  ADCO1 = HPUTAB(1,3)
000067 000      C  ELSE
000068 000      C  HAND-PLAYED ADCO1 IS BLANK THUS FASREC ADCO1 IS NOT CHG'D
000069 000      C  END-IF
000070 000      C  CHECK HAND-PLAYED ADCO2 FIELD FOR NON-BLANK
000071 000      C  IF (HPUTAB(1,4) .EQ. BLANK)  GOTO 1400
000072 000      C  THEN
000073 000      C  CHANGE FASREC ADCO2 TO HAND-PLAYED ADCO2
000074 000      C  ADCO2 = HPUTAB(1,4)
000075 000      C  ELSE
000076 000      C  HAND-PLAYED ADCO2 IS BLANK THUS FASREC ADCO2 IS NOT CHANGED
000077 000      C  END-IF
000078 000      C  CHECK HAND-PLAYED ADCO3 FIELD FOR NON-BLANK
000079 000      C  IF (HPUTAB(1,5) .EQ. BLANK)  GOTO 1500
000080 000      C  THEN
000081 000      C  CHANGE FASREC ADCO3 TO HAND-PLAYED ADCO3 AND SET RUDFLG

```

UNCLASSIFIED


```

000082      000      ADC03 = HPUTAB(1,5)
000083      000      RDEFLG = .TRUE.
000084      000      C      ELSE
000085      000      C      HAND-PLAYED ADC03 IS BLANK THUS NO-CHG TO FASREC
000086      000      C      END-IF
000087      000      C      SET POINTER TO ONE GREATER THAN LIMIT SO THAT WE EXIT FROM DOWNHILE
000088      000      1500      HPUTAB(1,6) = 1
000089      000      C      J = LIMIT + 1
000090      000      C      END-IF
000091      000      1900      CONTINUE
000092      000      C      GOTO 1000
000093      000      C      END-DO
000094      000      C
000095      000      C      END

```

END ELT.

WELT.L F5FS-CKLO

ELT007 RL1870 08/16-07:40:17-12.)

```

000001      000      C      C K L O - R O U T I N E
000002      000      C
000003      000      C
000004      000      C      RENE' R, PLOURDE
000005      000      C      USACAA/MRD JUN 1976
000006      000      C
000007      000      C      THIS ROUTINE IS USED TO DETERMINE WHETHER OR NOT A GIVEN FAS REC IS TO BE
000008      000      C      LOCKED OUT. IF ANY ONE OF THE FOLLOWING FIVE CONDITIONS IS TRUE THE LOCK-
000009      000      C      OUT-FLAG (LOFLAG) IS SET TO .TRUE.
000010      000      C
000011      000      C      1. ADC01 FROM FAS RECORD MATCHES AN ADC01 FROM THE LOCK-OUT TABLE (LOTAB)
000012      000      C      PROVIDED FOR BY USER
000013      000      C
000014      000      C      2. ALL TDA UNITS (SRC IS BLANK OR TPCO = 2 OR 3)
000015      000      C      3. SPLIT UNITS (UIC 1ST POS IS A W AND UIC 6TH POS IS > 0)
000016      000      C      4. DISPLAY ONLY UNITS
000017      000      C
000018      000      C      5. HAND-PLAYED COMPO'S
000019      000      C
000020      000      C      SUBROUTINE CKLO (ADC01,SRC15,TPCO,UIC,DSCMP,COMPO,ADC02,ADC03,
000021      000      C      SIZEC,SIZEC,LOTAB,HPCTAB,LOFLAG,WHY)
000022      002      C
000023      000      C      INTEGER LOTAB(100), HPCTAB(100)
000024      000      C      IMPLICIT INTEGER (A - Z)
000025      000      C      LOGICAL LOFLAG
000026      000      C
000027      000      C      DATA BLANK/1H /, TWO/1H2/, THREE/1H3/, W/1HW/, L/1HL/
000028      000      C      DATA DC/2HDC/, ZERO/4B/
000029      000      C
000030      000      C      WRITE (6,2) ADC01,SRC15,TPCO,UIC,DSCMP,COMPO,ADC02,
000031      000      C      * ADC03,SIZEC,SIZEC
000032      000      C      2 FORMAT(1X, SUB CKLO ADC01=,A1,1, SRC15=,A5,
000033      000      C      * TPCO=,A1,1, UIC=,A6,1, DSCMP=,A2,1, COMPO=,A1,
000034      000      C      * , ADC02=,A2,1, ADC03=,A3,1, SIZEC=,14,1, SIZEC=,14)
000035      000      C
000036      000      C

```

```
000037 000 LOFLAG = .FALSE.
000038 000 C
000039 000 C *** CHECK CONDITION 1 (ADCO1 FROM LOTAB) ***
000040 000 C
000041 000 I = 1
000042 000 C
000043 000 C DOWHILE (NOT EOF FOR LOTAB)
000044 000 C
000045 000 1000 IF (I.GT. SIZE) GOTO 2000
000046 000 C
000047 000 IF (ADCO1.NE. LOTAB(1)) GOTO 1100
000048 000 C THEN
000049 000 LOFLAG = .TRUE.
000050 001 WHY = 6HADCO1
000051 000 RETURN
000052 000 ELSE
000053 000 1100 I = I + 1
000054 000 C END-IF
000055 000 GOTO 1000
000056 000 C END-DO
000057 000 C
000058 000 C *** CHECK CONDITION 2 (TDA UNITS) ***
000059 000 C
000060 002 2000 IF (SRC15.NE. 0) GOTO 2100
000061 000 C THEN
000062 000 LOFLAG = .TRUE.
000063 001 WHY = 6HNOSRC
000064 000 RETURN
000065 000 C ELSE
000066 000 2100 IF (TYPCO.NE. TWO) GOTO 2200
000067 000 C THEN
000068 000 LOFLAG = .TRUE.
000069 001 WHY = 6HTYPCO2
000070 000 RETURN
000071 000 C ELSE
000072 000 2200 IF (TYPCO.NE. THREE) GOTO 2300
000073 000 C THEN
000074 000 LOFLAG = .TRUE.
000075 001 WHY = 6HTYPCO3
000076 000 RETURN
000077 000 C ELSE
000078 000 C NO-OP
000079 000 C END-IF
000080 000 2300 CONTINUE
000081 000 C
000082 000 2400 END-IF
000083 000 C CONTINUE
000084 000 2500 CONTINUE
000085 000 C
000086 000 C *** CHECK CONDITION 3 (SPLIT UNITS) ***
000087 000 C
000088 000 3000 DECODE (6,1,UIC) UICIST, UIC6TH
000089 000 1 FORMAT (A1,4X,A1)
000090 000 C
000091 000 IF (UICIST.NE. W) GOTO 3100
000092 000 C THEN
000093 000 IF (UIC6TH.LE. ZERO) GOTO 3100
```

```
000094 000 C THEN
000095 000 LOFLAG = .TRUE.
000096 001 WHY = 6MSPLIT
000097 000 RETURN
000098 000 C END-IF
000099 000 C END-IF
000100 000 3100 CONTINUE
000101 000 C
000102 000 C *** CHECK CONDITION 4 (DISPLAY ONLY UNITS) ***
000103 000 C
000104 000 4000 IF (DSCMP .EQ. DC) GOTO 4100
000105 000 C THEN
000106 000 LOFLAG = .TRUE.
000107 001 WHY = 6HDO
000108 001 RETURN
000109 000 C END-IF
000110 000 4100 CONTINUE
000111 000 C
000112 000 C *** CHECK CONDITION 5 (HAND-PLAYED COMPO'S) ***
000113 000 C
000114 000 5000 I = 1
000115 000 C
000116 000 C DOWHILE (NOT EOF FOR HPCTAB TABLE)
000117 000 C
000118 000 5100 IF (I .GT. SIZEC) GOTO 6000
000119 000 C
000120 000 IF (COMPO .NE. HPCTAB(I)) GOTO 5200
000121 000 C THEN
000122 000 LOFLAG = .TRUE.
000123 001 WHY = 6HCOMPO
000124 000 RETURN
000125 000 C ELSE
000126 000 5200 I = I + 1
000127 000 C END-IF
000128 000 5300 CONTINUE
000129 000 GOTO 5100
000130 000 C
000131 000 C END-DO
000132 000 C
000133 000 6000 CONTINUE
000134 000 C
000135 000 RETURN
000136 000 END

END ELT.
```

9ELT,L F5FS*CKSRC

ELT007 RL1870 08/16-07:40:19-(1.)

C K S R C - R U T I N E

RENE' R. PLOURDE
USACAA/MRD JUNE 1976THIS ROUTINE CHECKS SRC POSITIONS 8 AND 9 TO DETERMINE WHETHER OR NOT TO
CHANGE THEM TO 99 OR LEAVE UNCHANGED.

••UNCLASSIFIED••

FMA PROGRAM FSORT

```

000008 000 C
000009 000 C
000010 000 SUBROUTINE CKSRC (SRC15,SRC89)
000011 000 C
000012 000 IMPLICIT INTEGER (A = Z)
000013 000 C
000014 000 DATA A/IHA/, Z/IHZ/, N99/2H99/
000015 000 C
000016 000 DECODE (Z,I,SRC89) SRC8, SRC9
000017 000 I FORMAT(2A1)
000018 000 C
000019 000 WRITE (6,2) SRC15, SRC89
000020 000 2 FORMAT(' SUB CKSRC SRC15=',A5,'', SRC89=',A2)
000021 000 C
000022 000 C
000023 000 IF (SRC15 .EQ. 55018) GOTO 2000
000024 000 IF (SRC15 .EQ. 12067) GOTO 2000
000025 000 C
000026 000 IF (SRC8.GE.A .AND. SRC8.LE.Z) GOTO 1000
000027 000 C
000028 000 THEN SRC89 = N99
000029 000 GOTO 2000
000030 000 C
000031 000 ELSE
000032 000 1000 IF (SRC9.GE.A .AND. SRC9.LE.Z) GOTO 2000
000033 000 C
000034 000 THEN SRC89 = N99
000035 000 C
000036 000 ELSE NO=CHANGE
000037 000 C
000038 000 END=IF
000039 000 2000 CONTINUE
000040 000 RETURN
000041 000 END
END ELT.

```

BELT,L F5FS*CMCODE
ELT007 RL1870 08/16-07:40:19-(5,)

C M C O D E - R U T I N E

RENE' R. PLOURDE
USACAA/MRD JUNE 1976

THIS ROUTINE COMPUTES THE MCODE, (1, 2, OR 3) BY USING THE TABLES MCUTAB
C AND MCITAB, WHICH ARE PROVIDED FOR BY THE USER.

```

000008 001 C
000009 001 C SUBROUTINE CMCODE (TYPE,ADCO1,ROBCO,ADCO3,MCODE,COMPO,SIZE1,
000010 004 SIZEU,MCUTAB,MCITAB,MDDFLG)
000011 005 C
000012 001 C
000013 001 C IMPLICIT INTEGER (A=Z)
000014 001 C WRITE (6,99) TYPE,ADCO1,ROBCO,ADCO3,MCODE,SIZE1,SIZEU
000015 001 99 FORMAT('X,TYPE=',I1,ADCO1=',A1,ROBCO=',A3,ADCO3=',
000016 001 A3,MCODE=',A1,SIZE1=',I3,SIZEU=',I3)
000017 001 C INTEGER TYPE, MCUTAB(100,3)

```

FMA PROGRAM FSORT

••UNCLASSIFIED••


```
000018      LOGICAL RDOFLG
000019      DATA ASTER/IH/
000020      DATA ZERO/IHO/, ONE/IH1/, TWO/IH2/, THREE/IH3/
000021      DATA FOUR/IH4/, FIVE/IH5/
000022      C
000023      MDAY = MCOTAB(1)
000024      C
000025      MCODE = -1
000026      IF (COMPO.EQ.ONE.OR.COMPO.EQ.TWO.OR.COMPO.EQ.THREE)GOTO 100
000027      THEN
000028      C
000029      WE HAVE A NOTIONAL UNIT SET MCODE TO FIVE
000030      MCODE = FIVE
000031      RETURN
000032      END-IF
000033      C
000034      100 CONTINUE
000035      C
000036      IF (COMPO.NE.TWO .AND. COMPO.NE.THREE) GOTO 200
000037      THEN
000038      C
000039      WE HAVE EITHER A NATIONAL OR RESERVE UNIT SET MCODE TO FOUR
000040      MCODE = FOUR
000041      END-IF
000042      C
000043      200 CONTINUE
000044      C
000045      J = 2
000046      C
000047      IF (ADCO1.NE.MCOTAB(1)) GOTO 1500
000048      THEN
000049      C
000050      WE HAVE A MATCH FOR MCODE=U TABLE
000051      MCODE = ZERO
000052      IF (.NOT.RDOFLG) ADCU3=MDAY
000053      RDOFLG = .TRUE.
000054      RETURN
000055      END-IF
000056      C
000057      ELSE
000058      I = I + 1
000059      C
000060      1500 CONTINUE
000061      C
000062      IF (MCODE.EQ.FOUR) RETURN
000063      I = 1
000064      DECODE (J,1,ROBCO) R1, R2, R3
000065      I FORMAT (3A)
000066      C
000067      DOMHILE (NOT EOF FOR MCITAB TABLE)
000068      C
000069      3000 IF (I.GT.SIZE1) GOTO 4000
000070      DECODE (3,1,MCITAB(1,1)) T1, T2, T3
000071      IF (T1.NE.ASTER .AND. R1.NE.T1) GOTO 3100
000072      THEN
000073      C
000074      IF (T2.NE.ASTER .AND. R2.NE.T2) GOTO 3100
000075      THEN
000076      C
```

••UNCLASSIFIED••

FMA PROGRAM FSORT

```

000075 001 C
000076 001 C
000077 001 C
000078 001 C
000079 002 C
000080 001 C
000081 001 C
000082 001 C
000083 001 C
000084 001 C
000085 001 C
000086 001 C
000087 001 C
000088 001 C
000089 001 C
000090 001 C
000091 001 C
000092 001 C
000093 001 C
000094 001 C

IF (T3.NE.ASTER.AND.H3.NE.T3) GOTO 3100
THEN
    MCODE = ONE
    IF (.NOT.RDDFLG) ADC03=MCITAB(1,2+TYPE)
    RDDFLG = .TRUE.
    RETURN
END-IF

ELSE
    J = J + 1
    GOTO 3000
END-IF

END=DO
4000 CONTINUE

MCODE = THREE
RETURN
END

```

END ELT.

BELT.L FSFS=FCKUIC

```

ELT007 RL1870 08/16-07:40:21=(2,1)
000001 000 FUNCTION CKUIC (FASUIC,TABUIC)
000002 000 C
000003 000 C
000004 000 C
000005 000 C
000006 000 C
000007 000 C
000008 000 C
000009 000 C
000010 000 C
000011 000 C
000012 002 C
000013 000 C
000014 002 C
000015 000 C
000016 002 C
000017 000 C
000018 002 C
000019 000 C
000020 002 C
000021 000 C
000022 002 C
000023 000 C
000024 000 C
000025 000 C
000026 000 C
000027 000 C
000028 000 C
000029 000 C
000030 000 C

IMPLICIT INTEGER (A-Z)
DATA AST/IH*/

DECODE (6,1,FASUIC) F1, F2, F3, F4, F5, F6
DECODE (6,1,TABUIC) T1, T2, T3, T4, T5, T6
1 FORMAT (6A1)

WRITE (6,3) FASUIC, TABUIC
3 FORMAT (1H,'FUNCTION FCKUIC FASUIC=',A6,',', TABUIC=',A6)
IF (T1.NE.AST.AND.F1.NE.T1) GOTO 2
THEN
    IF (T2.NE.AST.AND.F2.NE.T2) GOTO 2
    THEN
        IF (T3.NE.AST.AND.F3.NE.T3) GOTO 2
        THEN
            IF (T4.NE.AST.AND.F4.NE.T4) GOTO 2
            THEN
                IF (T5.NE.AST.AND.F5.NE.T5) GOTO 2
                THEN
                    IF (T6.NE.AST.AND.F6.NE.T6) GOTO 2
                    THEN
                        CKUIC = 1
                        RETURN
                    END-IF
                END-IF
            END-IF
        END-IF
    END-IF
END-IF
END-IF
END-IF

```

••UNCLASSIFIED••

UNCLASSIFIED

FMA PROGRAM FSORT

```

000031 000 C      ELSE
000032 000      2      CKUIC = 0
000033 000      RETURN
000034 000 C      END-IF
000035 000      END

```

END ELT.

```

SELTL F5FS*MAIN
ELT007 RL1870 08/16-07:40:22-129,1
000001 025 C      F S O R T = R U T I N E
000002 025 C
000003 025 C
000004 025 C
000005 025 C
000006 025 C
000007 025 C
000008 025 C
000009 025 C
000010 025 C
000011 025 C
000012 025 C
000013 025 C
000014 025 C
000015 025 C
000016 025 C
000017 025 C
000018 025 C
000019 025 C
000020 025 C
000021 025 C
000022 025 C
000023 025 C
000024 025 C
000025 026 C
000026 025 C
000027 025 C
000028 025 C
000029 025 C
000030 025 C
000031 025 C
000032 025 C
000033 025 C
000034 025 C
000035 025 C
000036 025 C
000037 025 C
000038 025 C
000039 025 C
000040 025 C
000041 025 C
000042 025 C
000043 025 C
000044 025 C
000045 025 C

      THIS ROUTINE PROCESS THE FAS FILE. (FILE 2) A RECORD AT A TIME. EACH RECORD
      FROM FAS FILE IS WRITTEN TO ONE AND ONLY ONE OF THE OUTPUT FILES (CANDIDATE,
      HAND-PLAYES, LOCKOUT, ALT-THEATER, OR ABOVE-THE-LINE) WITH THE INPUT TABLES.
      (HAND-PLAYES-UNITS (HPUTAB), ALT-THEATER (ALTAB), LOCKOUT (LOTAB), MCODER,
      MCITAB), MCODE=0 (MCOTAB), AND HAND-PLAYED-COMPU (MPCTAB)) USED TO DETERMINE
      WHICH ONE OF THE OUTPUT FILES TO WRITE THE FAS RECORD TO.

      IMPLICIT INTEGER (A - Z)
      LOGICAL ERCODE, LOFLAG, MUDFLG, ALTFLG

      *** FILE DEFINITION ***

      DATA FAS/2/, CFILE/3/, LOFILE/20/, ATLFIL/21/, ALTFL/22/,
      * HPFILE/23/
      DATA FILEC/13/, FILELO/31/, FILATL/32/, FILALT/33/, FILEHP/34/

      C DEFINE VARIABLE ARRAYS TO INCLUDE THE SORT KEY USED BY THE SYSTEM SORT PROG

      INTEGER RECORD(13), FASREC(52), KEY(24)
      DATA KEY/1,1,42,0,0,1,2,13,72,0,0,2,9999,228*0/, BLANK/1H /
      DATA YES/3HYES/, NO/2HNO/, LEN/100/, N999/3H999/

      C THE ARRAY USES KEY(1) THRU KEY(6) TO DEFINE THE SORT KEY, WHERE

      C KEY(1) = 1 KEY BEGINS IN WORD 1 OF THE RECORD
      C KEY(2) = 1 KEY BEGINS IN BIT 1 OF THE WORD SPECIFIED BY KEY(1)
      C KEY(3) = 126 TOTAL NUMBER OF BITS USED TO DEFINE THE KEY
      C KEY(4) = 0 TYPE OF SORT, ALPHANUMERIC
      C KEY(5) = 0 TYPE OF ORDER, ASCENDING
      C KEY(6) = 1 OF THE TOTAL NUMBER OF KEYS THIS IS THE FIRST ONE
      C KEY(7) = 9999 INDICATES THAT THERE ARE NO MORE KEYS FOR THIS RECORD

      INTEGER PLIST(4), PLIST2(19), SIZES(6)

      C DEFINE INPUT TABLES AND THEIR MEANING
      INTEGER HPUTAB(100,6), LOTAB(100), ALTAB(100), MCITAB(100,3),
      * MPCTAB(100), MCOTAB(100)

      C HPUTAB(1,1) = UIC, HPUTAB(1,2) = RUBCO, HPUTAB(1,3) = ADCOL,

```

RENE' R. PLOURDE
USACAA JUNE 1976


```

000046 025 C HPUTAB(1,4) = ADCU2, HPUTAB(1,5) = ADCU3
000047 025 C LOTAB(1) = ADCU1, ALITAB(1) = ADCU1, HPCTAB(1) = COMPU
000048 025 C MCITAB(1,1) = ROBCO, MCITAB(1,2) = RDD=AL, MCITAB(1,3) = RDD=BTL
000049 025 C MCOTAB(1) = M=DAY, MCOTAB(1) = ADCU1 WHERE I>1
000050 025 C
000051 025 C *** OUTPUT STRUCTURE ***
000052 025 C
000053 025 C ALL FIVE OUTPUT FILES (CANDIDATE, HAND-PLAYED, LOCKOUT, ALT-THEATER, AND
000054 025 C ABOVE-THE-LINE) ARE BINARY FILES AND USE THE SAME RECORD STRUCTURE, WHERE
000055 025 C THE RECORD IS STRUCTURED AS FOLLOWS: (UIC1ST IS THE 1ST CHAR OF THE UIC)
000056 025 C
000057 025 C SRC15, SRCB9, COMPO, MCODE, UIC1ST, DAMPL, UIC, SEQ, ROBCO, ADCU1, ADCU2,
000058 025 C ADCU3, STNM, LOCCO, BRNCH, UNTDS, TPSN
000059 025 C
000060 025 C FOR CONVENIENCE WHEN ENCODING THE TWO LIST PLIST1 AND PLIST2 ARE USED
000061 025 C
000062 025 C EQUIVALENCE (PLIST1(1),SRC15), (PLIST1(2),SRCB9),
000063 025 C (PLIST1(3),COMPO), (PLIST1(4),MCODE)
000064 025 C
000065 025 C EQUIVALENCE (PLIST2(1),SRCB7), (PLIST2(2), SRCA),
000066 025 C (PLIST2(3), SRCB), (PLIST2(4),SRCCL),
000067 025 C (PLIST2(5),BRNCH), (PLIST2(6),UNTDL),
000068 025 C (PLIST2(7),UNTDL2), (PLIST2(8),UNTDL3),
000069 025 C (PLIST2(9),UNTDL4), (PLIST2(10),TPSN),
000070 025 C (PLIST2(11),ROBCO), (PLIST2(12),STNML),
000071 025 C (PLIST2(13),STNM2), (PLIST2(14),LOCCO),
000072 025 C (PLIST2(15),SAGR), (PLIST2(16),ADCO1),
000073 025 C (PLIST2(17),ADCO2), (PLIST2(18),ADCO3),
000074 025 C (PLIST2(19),SEQ)
000075 025 C
000076 025 C TLKOUT = 0
000077 025 C TALT = 0
000078 025 C TALT = 0
000079 025 C THPU = 0
000080 025 C TCAND = 0
000081 025 C SEQ = 0
000082 025 C
000083 025 C INITIALIZE SYSTEM SORT ROUTINE BY CALLING SOPEN3
000084 025 C
000085 025 C CALL SOPEN3 (100,13000,16,4,KEY)
000086 025 C
000087 025 C SET-UP-INPUT TABLES BY CALLING SUBROUTINE SUTAB
000088 025 C
000089 025 C 100 CALL SUTAB (HPUTAB,LOTAB,ALITAB,MCOTAB,MCITAB,HPCTAB,SIZES,
000090 025 C ERCODE)
000091 025 C
000092 025 C IF (ERCODE) STOP
000093 025 C
000094 025 C
000095 025 C DOWHILE (NOT EOF FOR FAS FILE)
000096 025 C
000097 025 C 1000 CONTINUE
000098 025 C
000099 025 C GET NEXT FASREC: DECODE IT, AND INCREMENT SEQ BY ONE
000100 025 C
000101 025 C READ (FAS,END=2000) FASREC
000102 025 C DECODE (100,1,FASREC) COMPO, UIC, BRNCH, DAMPL, DSCMP, LOCCO,

```



```

000103 025      * ADC01, ADC02, ADC03, ROBC0, SRC15, SRC67,
000104 025      * SRC89, SRCA, SRCC, STNM1, STNM2, TPSN, TYPCO, UNT01, UNT02,
000105 025      * UNT03, UNT04, SAGR
000106 025      C IF (SEQ.LE. 20) WRITE(6,7) ADC01, ADC02, ADC03
000107 025      7 FORMAT(' ADC01 = ', A1, ' ADC02 = ', A2, ' ADC03 = ', A3)
000108 025      C
000109 025      C CHECK FOR HAND-PLAYED-UNITS THEN CHECK FOR LOCKOUT-UNITS
000110 025      C
000111 025      CALL CKHPU (UIC, ROBCO, ADC01, ADC02, ADC03, SIZES(1),
000112 025      * MPUTAB, RUDFLG)
000113 025      CALL CKLU (ADC01, SRC15, TYPCO, UIC, USCMP, COMPO, ADC02, ADC03,
000114 025      * SIZES(2), SIZES(6), LUTAB, MPCTAB, LOFLAG, WHYLO)
000115 025      C
000116 025      IF (.NOT. LOFLAG) GOTO 1010
000117 025      C THEN
000118 025      C THIS UNIT IS A LOCKOUT UNIT THEREFORE WRITE IT TO THE LOFILE
000119 025      TLKOUT = TLKOUT + 1
000120 025      SEQ = TLKOUT
000121 025      C ENCODE (LEN, 2, RECORD) PLIST1, UIC, DAMPL, UIC, PLIST2
000122 025      C WRITE (LOFILE) RECORD
000123 025      C MCODE = BLANK
000124 025      WRITE (FILELO, 4) PLIST1, UIC, DAMPL, UIC, PLIST2, WHYLO
000125 025      GOTO 1990
000126 025      C ELSE
000127 025      C CHECK FOR ABOVE-THE-LINE (ATL) UNIT (TPSN < 20000) = (ATL)
000128 027      1010 CALL CKALT (ADC01, SIZES(3), ALTITAB, ALTFLG)
000129 027      IF (TPSN.GE. 20000) GOTO 1020
000130 025      C THEN
000131 025      C THE UNIT IS ABOVE-THE-LINE. THEREFORE CALCULATE THE MCODE
000132 025      C FOR THIS UNIT AND THEN WRITE THE RECORD TO ATFL.
000133 025      * CALL CMCODE (U, ADC01, ROBCO, ADC03, MCODE, COMPO,
000134 025      * SIZES(4), SIZES(5), MCUTAB, MCITAB, RUDFLG)
000135 025      TALT = TALT + 1
000136 027      FLGALT = NO
000137 027      IF (ALTFLG) FLGALT = YES
000138 025      SEQ = TALT
000139 025      HPFLAG = NO
000140 025      IF (RUDFLG) HPFLAG = YES
000141 025      WRITE (FILATL, 4) PLIST1, UIC, DAMPL, UIC, PLIST2,
000142 027      * HPFLAG, FLGALT
000143 025      GOTO 1980
000144 025      C ELSE
000145 025      C CHECK FOR ALTERNATE-THEATER (ALT)
000146 027      1020 IF (.NOT. ALTFLG) GOTO 1030
000147 025      C THEN
000148 025      C THIS UNIT IS PART OF THE ALT-THEATER THEREFORE
000149 025      C WRITE IT TO THE ATFL FILE
000150 025      TALT = TALT + 1
000151 025      SEQ = TALT
000152 029      IF (.NOT. RUDFLG) ADC03 = N999
000153 025      C ENCODE ( LEN, 2, RECORD) PLIST1, UIC, DAMPL,
000154 025      C * UIC, PLIST2
000155 025      C WRITE (ATFIL) RECORD
000156 025      C MCODE = BLANK
000157 025      WRITE (FILALT, 4) PLIST1, UIC, DAMPL, UIC,
000158 025      * PLIST2
000159 025      GOTO 1960

```



```
000217 025 CSEQ = CSEQ + 1
000218 025 C WRITE (CFIL) RECORD
000219 027 DECODE (LEN,2,RECORD) PLIST1, UIC1, DAMPL, UIC, PLIST2, SAVE89
000220 025 SEQ = CSEQ
000221 027 WRITE (FILEC,4) PLIST1, UIC1, DAMPL, UIC, PLIST2, SAVE89
000222 025 GOTO 3100
000223 025 C END=00
000224 025 C
000225 025 4000 CONTINUE
000226 025 I = 1
000227 025 C
000228 025 C CHECK TO SEE IF ANY HAND-PLAYED-UNITS FROM HPUTAB WERE NOT PLAYED
000229 025 C
000230 025 C DOWHILE (NOT EOF FOR HPUTAB)
000231 025 C
000232 025 4100 IF (I.GT. SIZE(1)) GOTO 5000
000233 025 C
000234 025 IF (HPUTAB(I,6) .NE. 0) GOTO 4200
000235 025 C THEN
000236 025 WRITE (6,3) I, (HPUTAB(I,J),J=1,6)
000237 025 C END-IF
000238 025 4200 CONTINUE
000239 025 I = I + 1
000240 025 GOTO 4100
000241 025 C END=00
000242 025 5000 CONTINUE
000243 025 C
000244 025 C PRINT ACCUMULATED TOTAL AND SUB-TOTALS THEN STOP
000245 025 C
000246 025 TOTAL = TLKOUT + TATL + TALT + THPU + ICAND
000247 025 WRITE (6,50) TLKOUT, TATL, TALT, THPU, ICAND, TOTAL
000248 025 50 FORMAT (1H1, '*** T O T A L S ***//
000249 025 * LOCKOUT =',18/
000250 025 * ABOVE-THE-LINE =',18/
000251 025 * ALT-THEATER =',18/
000252 025 * HAND-PLAYED-UNITS =',18/
000253 025 * MATCH-CANDIDATES =',18//
000254 025 * TOTAL =',18)
000255 025 C
000256 025 1 FORMAT (T3,A1,T4,A6,T44,A2,T58,A5,T71,A2,T88,A3,T19,A1,T20,A2,
000257 025 * T22,A3,T30,A3,T34,T5,ZA2,ZA1,T54,A6,T60,A3,T80,T5,T86,A1/
000258 025 T4,A3,T7,A6,T13,A6,T14,A6,T93,A5)
000259 025 2 FORMAT (J5,A2,ZA1,A5,A6,A2,ZA1,A2,A3,ZA6,J5,A3,A6,ZA3,A5,
000260 025 * A1,A2,A3,16,A6)
000261 025 3 FORMAT (1H0, 'HAND-PLAYED-UNIT ',14, ' WAS NOT USED UIC = ',
000262 025 * A6, ' , RUBCO = ',A3, ' , ADC01 = ',A1, ' , ADC02 = ',A2, ' , ADC03 = ',
000263 025 * A3, ' , FLAG = ',11)
000264 025 4 FORMAT (1X,J5,A2,ZA1,A5,A6,A2,ZA1,A2,A3,ZA6,J5,A3,A6,ZA3,A5,
000265 027 * A1,A2,A3,16,A6,A3)
000266 025 5 FORMAT(A1)
000267 025 ENDFILE FILELO
000268 025 ENDFILE FILATL
000269 025 ENDFILE FILALT
000270 025 ENDFILE FILEHP
000271 025 ENDFILE FILEC
000272 025 STOP
000273 025 END
```


END ELT.

```

BELT,L  FSFS,MODUIC
ELT007  RL1870 08/16-07:40:24-10.)
000001 000 C
000002 000 C M O D U I C
000003 000 C
000004 000 C
000005 000 C
000006 000 C
000007 000 C SUBROUTINE MODUIC (COMPO,UICI)
000008 000 C
000009 000 C IMPLICIT INTEGER (A - Z)
000010 000 C
000011 000 C DATA K/1HK/, X/1HX/, W/1HW/
000012 000 C DATA ONE/1H1/, TWO/1H2/, THREE/1H3/
000013 000 C
000014 000 C IF (COMPO.NE.ONE) GOTO 50
000015 000 C THEN
000016 000 C IF (UICI.NE.W) GOTO 10
000017 000 C THEN
000018 000 C UICI = ONE
000019 000 C GOTO 40
000020 000 C ELSE
000021 000 C 10 IF (UICI.NE.X) GOTO 20
000022 000 C THEN
000023 000 C UICI = TWO
000024 000 C GOTO 30
000025 000 C ELSE
000026 000 C 20 ELSE UICI = THREE
000027 000 C END-IF
000028 000 C CONTINUE
000029 000 C 30 END-IF
000030 000 C CONTINUE
000031 000 C 40 GOTO 120
000032 000 C ELSE
000033 000 C 50 IF (COMPO.NE.TWO.AND.COMPO.NE.THREE) GOTO 100
000034 000 C THEN
000035 000 C IF (UICI.NE.W) GOTO 60
000036 000 C THEN
000037 000 C UICI = ONE
000038 000 C GOTO 90
000039 000 C ELSE
000040 000 C 60 IF (UICI.NE.K) GOTO 70
000041 000 C THEN
000042 000 C UICI = TWO
000043 000 C GOTO 80
000044 000 C ELSE
000045 000 C 70 ELSE UICI = THREE
000046 000 C END-IF
000047 000 C 80 CONTINUE
000048 000 C END-IF
000049 000 C CONTINUE
000050 000 C 90 GOTO 110
000051 000 C ELSE
```

RENE' R. PLOURDE
USACAA/MRU JUNE 1976


```
000043 000      1 FORMAT (A6,1X,A3,1X,A1,1X,A2,1X,A3)
000044 000      11 FORMAT (1H,14,3X,A6,3X,A3,7X,A1,6X,A2,6X,A3)
000045 000      C
000046 000      C PROCESS ERR FOR TABLE HPUTAB
000047 000      C
000048 000      C
000049 000      51 FORMAT (1H0,1000 FORMAT ERR ***)
000050 000      C
000051 000      2000 CONTINUE
000052 000      SIZE(1) = N - 1
000053 000      IF (SIZE(1) .LE. LIMIT1) GOTO 2050
000054 000      ERCODE = .TRUE.
000055 000      WRITE (6,61)
000056 000      61 FORMAT (1H0,1000 TO MANY INPUT RECORDS ***)
000057 000      2050 CONTINUE
000058 000      WRITE (6,102)
000059 000      102 FORMAT (1H1,1000 LOCKOUT TABLE ***/1H,1' SEQ ADCOL1')
000060 000      N = 0
000061 000      C SET-UP LOCKOUT TABLE (LOTAB)
000062 000      C
000063 000      C
000064 000      C DOWHILE (NOT EOF AND NO ERKS)
000065 000      C
000066 000      2100 CONTINUE
000067 000      N = N + 1
000068 000      C
000069 000      READ (LOUNIT,2,END=3000) LOTAB(N)
000070 000      WRITE (6,12) N, LOTAB(N)
000071 000      GOTO 2100
000072 000      C
000073 000      2 FORMAT (A1)
000074 000      12 FORMAT (1H,14,5X,A1)
000075 000      C
000076 000      C PROCESS ERR FOR TABLE LOTAB
000077 000      C
000078 000      C
000079 000      3000 CONTINUE
000080 000      SIZE(2) = N - 1
000081 000      IF (SIZE(2) .LE. LIMIT2) GOTO 3050
000082 000      ERCODE = .TRUE.
000083 000      WRITE (6,61)
000084 000      3050 CONTINUE
000085 000      WRITE (6,103)
000086 000      103 FORMAT (1H1,1000 ALTERNATE-THEATER-TABLE ***/1H,1'
000087 000      * SEQ ADCOL1')
000088 000      N = 0
000089 000      C SET-UP ALTERNATE-THEATER TABLE (ALTAB)
000090 000      C
000091 000      C
000092 000      C DOWHILE (NOT EOF AND NO ERKS)
000093 000      C
000094 000      3100 CONTINUE
000095 000      N = N + 1
000096 000      C
000097 000      READ (ALH,3,END=4000) ALTAB(N)
000098 000      WRITE (6,13) N, ALTAB(N)
000099 000      GOTO 3100
```

```
000100 000 3 FORMAT (A1)
000101 000 13 FORMAT (1H ,14,5X,A1)
000102 000 C
000103 000 C
000104 000 C PROCESS ERR FOR ALTERNATE-THEATER INPUT TABLE (ALITAB)
000105 000 C
000106 000 C
000107 000 4000 CONTINUE
000108 000 SIZE(3) = N - 1
000109 000 IF (SIZE(3) .LE. LIMIT3) GOTO 4050
000110 000 ERCODE = .TRUE.
000111 000 WRITE (6,61)
000112 000 4050 CONTINUE
000113 000 WRITE (6,104)
000114 000 104 FORMAT (1H1,*,** MCODE = 1 TABLE ***,//1H ,
000115 000 * , SEQ ROBCO RDD/ATL RDD/BTL//)
000116 000 N = 0
000117 000 C
000118 000 C SET-UP MCODE = 1 TABLE (MCITAB)
000119 000 C
000120 000 C DOWHILE (NOT EOF AND NO ERRS OCCUR)
000121 000 C
000122 000 4100 CONTINUE
000123 000 N = N + 1
000124 000 C
000125 000 READ (MCUNE,*,END=5000) (MCITAB(N,J),J=1,3)
000126 000 WRITE (6,14) N, (MCITAB(N,J),J=1,3)
000127 000 GOTO 4100
000128 000 C
000129 000 4 FORMAT (A3,2X,A3,2X,A3)
000130 000 14 FORMAT (1H ,14,3X,A3,7X,A3,A3)
000131 000 C
000132 000 C PROCESS ERR FOR MCODE = 1 TABLE (MCITAB)
000133 000 C
000134 000 C
000135 000 5000 CONTINUE
000136 000 SIZE(4) = N - 1
000137 000 IF (SIZE(4) .LE. LIMIT4) GOTO 5050
000138 000 ERCODE = .TRUE.
000139 000 WRITE (6,61)
000140 000 5050 CONTINUE
000141 000 WRITE (6,105)
000142 000 105 FORMAT (1H1,*,** MCODE = 0 TABLE ***,//1H ,
000143 000 * , SEQ COMPO//)
000144 000 N = 0
000145 000 C
000146 000 C SET-UP MCODE = 0 TABLE (MCOTAB)
000147 000 C
000148 000 5100 CONTINUE
000149 000 N = N + 1
000150 000 C
000151 000 READ (MCZERO,5,END=6000) MCOTAB(N)
000152 000 WRITE (6,15) N, MCOTAB(N)
000153 000 GOTO 5100
000154 000 C
000155 000 5 FORMAT (A6)
000156 000 15 FORMAT (1H ,14,5X,A6)
```

```
000157 000 C
000158 000 C PROCESS ERR FOR MCODE = 0 TABLE (MCOTAB)
000159 000 C
000160 000 C
000161 000 6000 CONTINUE
000162 000 SIZE(5) = N - 1
000163 000 IF (SIZE(5) * LE * LIMITS) GOTO 6050
000164 000 ERCODE = .TRUE.
000165 000 WRITE (6,61)
000166 000 6050 CONTINUE
000167 000 WRITE (6,106)
000168 000 106 FORMAT (1H1,10000 HAND-PLAYED-COMPOS-TABLE ****//1H ,
* , SEQ COMPO//)
000169 000
000170 000 N = 0
000171 000 C SET-UP HAND-PLAYED-COMPOS TABLE (HPCTAB)
000172 000 C
000173 000 C DOWHILE (NOT EOF AND NO ERRORS EXIST)
000174 000 C
000175 000 6100 CONTINUE
000176 000 N = N + 1
000177 000 C
000178 000 READ (HPCOMP,6,END=7000) HPCTAB(N)
000179 000 WRITE (6,16) N, HPCTAB(N)
000180 000 GOTO 6100
000181 000 C
000182 000 6 FORMAT (A6)
000183 000 16 FORMAT (1H ,14,3X,A6)
000184 000 C
000185 000 C PROCESS ERR FOR HAND-PLAYED-COMPO TABLE (HPCTAB)
000186 000 C
000187 000 C
000188 000 C
000189 000 7000 CONTINUE
000190 000 SIZE(6) = N - 1
000191 000 IF (SIZE(6) * LE * LIMIT6) GOTO 8000
000192 000 ERCODE = .TRUE.
000193 000 WRITE (6,61)
000194 000 8000 CONTINUE
000195 000 CALL CLOSE(HPUNIT,1)
000196 000 CALL CLOSE(ALTH,1)
000197 000 CALL CLOSE(MCONE,1)
000198 000 CALL CLOSE(MCZERO,1)
000199 000 CALL CLOSE(HPCOMP,1)
000200 000 C
000201 000 C TIME TO RETURN
000202 000 C
000203 000 RETURN
000204 000 END
```

END ELT.

ENDG FMA PROGRAM FSORT

UNCLASSIFIED .L.U


```
BELT.L F5RS*CKSRC
ELT007 RL1870 08/16-07:40:27-(2,1) C K S R C - R O U T I N E
000001 000 C
000002 000 C
000003 000 C
000004 000 C
000005 000 C
000006 000 C
000007 000 C
000008 000 C
000009 000 C
000010 000 C
000011 000 C
000012 000 C
000013 000 C
000014 000 C
000015 000 C
000016 000 C
000017 000 C
000018 000 C
000019 001 C
000020 000 C
000021 000 C
000022 000 C
000023 002 C
000024 002 C
000025 000 C
000026 000 C
000027 000 C
000028 000 C
000029 000 C
000030 000 C
000031 000 C
000032 000 C
000033 000 C
000034 000 C
000035 000 C
000036 000 C
000037 000 C
000038 000 C
000039 000 C
000040 000 C

THIS ROUTINE CHECKS SRC POSITIONS 8 AND 9 TO DETERMINE WHETHER OR NOT TO
CHANGE THEM TO 99 OR LEAVE UNCHANGED.

SUBROUTINE CKSRC (SRC15,SRC89)
IMPLICIT INTEGER (A-Z)
DATA A/IHA/, Z/1HZ/, N99/2499/
DECODE (Z,I,SRC89) SRC8, SRC9
1 FORMAT(2A1)
IF (SRC15.EQ.55018) GOTO 2000
IF (SRC15.EQ.12067) GOTO 2000
IF (SRC8.GE.A.AND. SRC8.LE.Z) GOTO 1000
THEN SRC89 = N99
GOTO 2000
ELSE
1000 IF (SRC9.GE.A.AND. SRC9.LE.Z) GOTO 2000
THEN SRC89 = N99
ELSE
NO-CHANGE
END-IF
END-IF
2000 CONTINUE
RETURN
END

END ELT.
```

RENE' R. PLOURDE
USACAA/MRD JUNE 1976

```
BELT.L F5RS*RSORT
ELT007 RL1870 08/16-07:40:27-(2,1) R S O R T - R O U T I N E
```

```
000001 000 C
000002 000 C
000003 000 C
000004 000 C
000005 000 C
000006 000 C
000007 000 C
000008 000 C

THIS ROUTINE EXAMINES THE FASTAL REQUIREMENT FILE AND USES THE HAND-PLAYED-
C UIN FILE, (PROVIDED FOR BY THE USER) TO BUILD A REQUIREMENT FILE.
```

RENE' R. PLOURDE
USACAA/MRD JUNE 1976

```
000009 000 IMPLICIT INTEGER (A-Z)
000010 000 INTEGER RECORD(13), LIST(23), KEY(241), UNTAB(100,3)
000011 000 LOGICAL UINFLG
000012 000 EQUIVALENCE (LIST(1),SRC15), (LIST(2),SRC89), (LIST(3),TMPD),
000013 000 (LIST(4),SRC67), (LIST(5),SRCA), (LIST(6),SRCB),
000014 000 (LIST(7),SRCC), (LIST(8),UNT01), (LIST(9),UNT02),
000015 000 (LIST(10),UNT03), (LIST(11),UNT04), (LIST(12),RULE),
000016 000 (LIST(13),LRE), (LIST(14),COMB), (LIST(15),ROUND),
000017 000 (LIST(16),TPSN), (LIST(17),SOFF), (LIST(18),SWOF),
000018 000 (LIST(19),SEN), (LIST(20),SAGR), (LIST(21),UIN),
000019 000 (LIST(22),BRNCH), (LIST(23),STR)
000020 000 DATA LEN/100/, KEY/1,1,54,0,0,1,9999,234*0/
000021 000 DATA FPUIN/1/, FILREG/2/, FILTDA/3/, FASTAL/14/,
000022 000 FUINHP/11/, REWFIL/12/, TDAFIL/13/, HPUINT/15/
000023 000 C
000024 000 C OPEN SORT ROUTINE
000025 000 C
000026 000 CALL SOPEN3 ($1000,$4000,13,2,KEY)
000027 000 C
000028 000 C READ FIRST TIME PERIOD TO ACCEPT REQUIREMENTS
000029 000 1000 READ (HPUINT,4) FIRSTP
000030 000 WRITE (6,13) FIRSTP
000031 000 C
000032 000 C INPUT HAND PLAYED UIN'S
000033 000 C
000034 000 C WRITE (6,6)
000035 000 C SIZE = 1
000036 000 C
000037 000 C DOWHILE (NOT=EOF=FOR=HPUINT TABLE)
000038 000 C
000039 000 1050 READ (HPUINT,4,END=1100) UNTAB(SIZE,1), UNTAB(SIZE,2)
000040 000 WRITE (6,5) SIZE, UNTAB(SIZE,1), UNTAB(SIZE,2)
000041 000 SIZE = SIZE + 1
000042 000 GOTO 1050
000043 000 C END=00
000044 000 C
000045 000 1100 CONTINUE
000046 000 SIZE = SIZE - 1
000047 000 SEQ = 1
000048 000 C
000049 000 C DOWHILE (NOT=EOF=FASTAL=FILE)
000050 000 C
000051 000 2000 READ (FASTAL,1,END=3000) UIN, SRC15, SRC67, SRC89, SRCA, SRCB,
000052 000 (SRC, BRNCH, UNTD1, UNTD2, UNTD3, UNTD4, STR, TMPD, RULE,
000053 000 (LRE, COMB, ROUND, TPSN, SOFF, SWOF, SENL, SAGR
000054 000 C IF (SEQ.LE.100) WRITE(6,3) LIST
000055 000 SEQ = SEQ + 1
000056 000 C
000057 000 C CHECK FOR TDA UIN'S (SRC15 IS BLANK)
000058 000 C
000059 000 C IF (SRC15.EQ.0) GOTO 2300
000060 000 C THEN
000061 000 C CHECK FOR HAND-PLAYED UIN
000062 000 UINFLG = .FALSE.
000063 000 IF (TMPD.LT. FIRSTP) UINFLG = .TRUE.
000064 000 I = 1
000065 000 C
```

```
000066 000 C DOWHILE (NOT END-OF-TABLE UINTAB)
000067 000 C
000068 000 2100 IF (UINFLG *OR* 1 *GT* SIZE) GOTO 2199
000069 000 C
000070 000 IF (UIN *NE* UINTAB(1,1)) GOTO 2150
000071 000 IF (TMPD*LT* UINTAB(1,2)) GOTO 2150
000072 000 C THEN
000073 000 UINFLG = *TRUE*
000074 000 UINTAB(1,3) = UINTAB(1,3) + 1
000075 000 GOTO 2160
000076 000 C ELSE
000077 000 2150 I = I + 1
000078 000 C
000079 000 2160 END-IF
000080 000 CONTINUE
000081 000 GOTO 2100
000082 000 C END-DO
000083 000 2199 IF (*NOT*UINFLG) GOTO 2200
000084 000 C THEN
000085 000 C THIS RECORD IS A HAND-PLAYED-UIN UNIT
000086 000 C IF(SEQ*LE*101) WRITE(6,8)
000087 000 C ENCODE (LEN,2,RECORD) LIST
000088 000 C WRITE (FHPUIN) LIST
000089 000 C WRITE (FUINHP,3) LIST
000090 000 C THPUIN = THPUIN + 1
000091 000 C GOTO 2250
000092 000 C ELSE
000093 000 C WE HAVE A REQ RECORD
000094 000 2200 SAVE89 = SRC89
000095 000 C CALL CKSRC (SRC15, SRC89)
000096 000 C IF(SEQ*LE*101) WRITE(6,9)
000097 000 C ENCODE (LEN,2,RECORD) LIST, SAVE89
000098 000 C CALL SRREL (RECORD,13)
000099 000 C TREQ = TREQ + 1
000100 000 C
000101 000 2250 END-IF
000102 000 C CONTINUE
000103 000 C GOTO 2400
000104 000 C ELSE
000105 000 C WE HAVE A TDA UNIT
000106 000 2300 CONTINUE
000107 000 C IF(SEQ*LE*101) WRITE(6,10)
000108 000 C ENCODE (32,2,RECORD) LIST
000109 000 C WRITE (TDAFIL,3) LIST
000110 000 C TDA = TDA + 1
000111 000 C
000112 000 2400 END-IF
000113 000 C CONTINUE
000114 000 C GOTO 2000
000115 000 C END-DO
000116 000 C
000117 000 3000 CONTINUE
000118 000 C
000119 000 C TIME TO SORT REQUIREMENT'S FILE
000120 000 C CALL SSORT
000121 000 C
000122 000 4000 CONTINUE
000123 000 C SEQ = 0
```



```
000123 000 4100 CONTINUE
000124 000 SEQ = SEQ + 1
000125 000 CALL SRRET (RECORD,MAX,$5000)
000126 001 DECODE (LEN,2,RECORD) LIST, SAVEB9
000127 001 WRITE (REQFIL,3) LIST, SEQ, SAVEB9
000128 000 GOTO 4100
000129 000 C
000130 000 C CHECK UNTAB TO SEE IF ANY HAND-PLAYED UIN'S WERE NOT USED
000131 000 C
000132 000 5000 CONTINUE
000133 000 I = 1
000134 000 WRITE (6,12)
000135 000 C
000136 000 C DOWHILE (NOT END OF UNTAB TABLE)
000137 000 C
000138 000 5100 IF (I.GT. SIZE) GOTO 6000
000139 000 C
000140 000 WRITE (6,11) I, (UNTAB(I,J),J=1,3)
000141 000 5200 I = I + 1
000142 000 GOTO 5100
000143 000 C END-DO
000144 000 C
000145 000 6000 CONTINUE
000146 000 C
000147 000 C COMPUTE TOTAL AND PRINT
000148 000 C
000149 000 TOTAL = THPUIN + TTDA + TREQ
000150 000 WRITE (6,7) THPUIN, TTDA, TREQ, TOTAL
000151 000 1 FORMAT(13,2X,15,2A2,3A1,2X,A2,1X,A3,3A6,12X,13,1X,12,2X,A4,
000152 000 * 3(1X,A1),1X,15,4(1X,A5))
000153 001 2 FORMAT(15,A2,12,A2,3A1,A3,3A6,A4,3A1,15,4A5,13,A2,13,A3)
000154 000 3 FORMAT(1X,15,1X,A2,1X,12,1X,A2,3(1X,A1),1X,A3,3(1X,A6),1X,A4,
000155 002 * 3(1X,A1),1X,15,4(1X,A5),1X,13,1X,A2,1X,13,16,A3)
000156 000 4 FORMAT(13,1X,12)
000157 000 5 FORMAT(11,14,3X,13,3X,12)
000158 000 6 FORMAT(11,14,3X,13,3X,12)
000159 000 7 FORMAT(11,14,3X,13,3X,12)
000160 000 * 1 HAND-PLAYED-UIN =',18/
000161 000 * 1 TDA-UNITS =',18/
000162 000 * 1 REQUIREMENTS =',18//
000163 000 * 1 TOTAL =',18)
000164 000 8 FORMAT(1,14,3X,13,3X,12)
000165 000 9 FORMAT(1,14,3X,13,3X,12)
000166 000 10 FORMAT(1,14,3X,13,3X,12)
000167 000 11 FORMAT(11,14,3X,13,3X,12,3X,16)
000168 000 12 FORMAT(11,14,3X,13,3X,12,3X,16)
000169 000 13 FORMAT(11,14,3X,13,3X,12,3X,16)
000170 000 STOP
000171 000 END
END ELT.
```

SHDG FMA PROGRAM MATCH

UNCLASSIFIED L,0


```
SELT,L FSFM,PRINT
ELT007 RL1870 08/16-07:40:29-(10,1)
000001 006 C
000002 006 C P R I N T - R O U T I N E
000003 006 C
000004 006 C SUBROUTINE PRT (LINE,TYPE)
000005 006 C
000006 006 C IMPLICIT INTEGER (A-Z)
000007 006 C COMMON /FASTAL/ MBRNCH,MSRC15,MSRC67,MSRC69,MSRCA,MSRCB,MSRCC,
000008 006 C RTPSN,RUNT01,RUNT02,MSRC67,MSRC69,MSRCA,MSRCB,MSRCC,NAUIC,
000009 006 C RSENL,RSAGR,UIIN,RULE,TPD,LRE,UMBR,OLDSRC,NAUIC,
000010 009 C RSTR,RESEQ,MSTNM1,MSTNM2,LUCCO,MCOMPO,SFMB9
000011 006 C COMMON /MCAND/ MSRC15,MSRC67,MSRC69,MSRCA,MSRCB,MSRCC,COMPO,
000012 006 C MC,UICIST,DAMPL,UIC,MUNT01,MUNT02,MUNT03,MUNT04,
000013 009 C MCEQ,RTPSN,MBRNCH,MSAGR,LUCCO,MSTNM1,MSTNM2,ROBCO,SFMB9
000014 006 C DATA RESULT/19/
000015 006 C
000016 006 C *** CHECK TO SEE IF THE HEADER NEEDS TO BE PRINTED ***
000017 006 C
000018 008 C IF (LINE.LT. 54) GOTO 500
000019 006 C THEN
000020 006 C PRINT THE HEADER
000021 006 C WRITE (RESULT,1)
000022 006 C LINE = 1
000023 006 C GOTO 600
000024 006 C ELSE
000025 006 C INCREMENT LINE COUNTER
000026 006 C 500 LINE = LINE + 1
000027 006 C END-IF
000028 006 C
000029 006 C 600 CONTINUE
000030 006 C
000031 006 C *** CHECK FOR SHORT-FALL (TYPE = 1) ***
000032 006 C
000033 006 C IF (TYPE.NE. 1) GOTO 1000
000034 006 C THEN
000035 006 C PRINT ONE EACH SHORT-FALL LINE
000036 009 C WRITE (RESULT,2) MSRC15,SFMB9,MBRNCH,RUNT01,RUNT02,
000037 007 C RUNT03,RUNT04,RCOMPO,UIIC,RTPSN,UIIN,TPD,RESEQ,
000038 007 C RSTNM1,RSTNM2,LUCCO,RSTR,MSAGR
000039 006 C RETURN
000040 006 C END-IF
000041 006 C
000042 006 C *** CHECK FOR A MATCH (TYPE = 2) **
000043 006 C
000044 006 C 1000 IF (TYPE.NE. 2) GOTO 2000
000045 006 C THEN
000046 006 C PRINT ONE EACH MATCH LINE
000047 010 C WRITE (RESULT,3) MSRC15,SFMB9,MSRC15,SFMB9,MBRNCH,
000048 006 C MUNT01,MUNT02,MUNT03,MUNT04,COMPO,UIC,
000049 006 C RTPSN,UIIN,TPD,RESEQ,MC,ROBCO,MSTNM1,
000050 006 C MSTNM2,LUCCO,RSTR,MSAGR,MCSEQ
000051 006 C RETURN
000052 006 C END-IF
000053 006 C
000054 006 C *** CHECK FOR AN OVERAGE (TYPE = 3) ***
000055 006 C
```

```

000056 006 2000 IF (TYPE.NE. 3) GOTO 3000
000057 006 C THEN
000058 006 C PRINT ONE EACH OVERAGE LINE
000059 009 WRITE (RESULT,4) MSRC15,SFMB89,MBRNCH,MUNTD1,MUNTD2,
000060 006 MUNT03,MUNT04,COMPO,UIC,MTPSN,MC,
000061 006 ROBCU,MSTNM1,MSTNM2,LOCCO,MSAGR,MCSE9
000062 006 RETURN
000063 006 C END-IF
000064 006 C
000065 006 C *** ERROR TYPE.NE. 1, 2, OR 3 ***
000066 006 3000 WRITE (6,99) TYPE
000067 006 99 FORMAT (' *** ERROR TYPE =',I2)
000068 006 1 FORMAT (1H1,'REQ-SRC FORCE-SRC BRNCH LEVEL UNITS',I4X,
000069 006 ' * COMPO UIC TPN UN TP RSEQ MC ROBCO STNM',5X,
000070 006 ' * LOCCO RSTR FSTR FSEQ',//)
000071 006 2 FORMAT (1X,J5,1X,A2,14X,A2,3X,A3,2X,3A6,3X,A1,3X,A6,1X,J5,
000072 006 ' 1X,13,1X,12,16,10X,A6,A3,2X,A3,2X,J5,1X,A5)
000073 006 3 FORMAT (1X,J5,1X,A2,2X,J5,1X,A2,4X,A2,3X,A3,2X,3A6,3X,A1,
000074 006 ' 3X,A6,1X,J5,1X,13,1X,12,1X,15,1X,A1,3X,A3,2X,A6,A3,
000075 006 ' 2X,A3,2X,J5,1X,A5,1X,15)
000076 006 4 FORMAT (11X,J5,1X,A2,4X,A2,3X,A3,2X,3A6,3X,A1,3X,A6,1X,15,
000077 006 ' 14X,A1,3X,A3,2X,A6,A3,2X,A3,8X,A5,1X,15)
000078 006 END

```

END ELT.

```

BELT,L FSFM,SHORT
ELT007 RL1870 08/16-07:40:30-(23,1)
000001 019 C S H O R T - R O U T I N E
000002 019 C
000003 019 C
000004 019 C SUBROUTINE SHORT (NEWSRC)
000005 019 C
000006 019 C IMPLICIT INTEGER (A - Z)
000007 019 COMMON /TABLES/ NOTAB(20), PTAB(4,23), KUOTAB(7,10),
000008 019 STP(6,10), KNTRDU(10)
000009 019 COMMON /REGREC/ RECORD(25)
000010 019 COMMON /FASTAL/ RBRNCH,MSRC15,RSRC67,MSRC89,MSRCA,RSRCB,RSRCC,
000011 019 RTPSN,RUNTD1,RUNTD2,RUNTD3,RUNTD4,R5OFF,RSWOF,
000012 019 RSENL,RSAGR,UIIN,RULE,TPD,LRE,UNMBR,OLDSRC,NAUIC,
000013 019 RSTR,REGSEQ,RSTNM1,RSTNM2,RUIC,LOCCO,COMPO,SFMB9
000014 019 COMMON /MCAND/ MSRC15,MSRC67,MSRC89,MSRCA,MSRCB,MSRCC,COMPO,
000015 019 MC,UICIST,DAMPL,UIC,MUNTD1,MUNTD2,MUNTD3,MUNTD4,
000016 019 MCSE9,MTPSN,MBRNCH,MSAGR,LOCCO,MSTNM1,MSTNM2,ROBCO,SFMB9
000017 019 C
000018 019 INTEGER TINDEX(18), RSHORT(47)
000019 019 DATA TINDEX/1,4,5,6,9,14,15,16,17,18,19,20,21,30,31,32,34,2/
000020 019 DATA ONE/1H1/, FIVE/1H5/, B/1H8/, BLANK/1H /, ASTER/6H####/,
000021 021 DATA MCODE9/1H9/,DAMPL9/5H99999/,ZERO/1H0/,RSHORT/47*6H0000000/
000022 019 C
000023 019 C *** TRANSFER NOTIONAL TABLE INPUT
000024 019 C
000025 019 C I = 1
000026 019 C
000027 019 C DO WHILE (I.LE. 18)

```

```
000028 019 C 1000 IF (I.GT.18) GOTO 1100
000029 019 C
000030 019 C
000031 019 K = INDEX(I)
000032 019 RSHORT(K) = NOTAB(I)
000033 019 I = I + 1
000034 019 GOTO 1000
000035 019 C END=DO
000036 019 C
000037 019 C *** TRANSFER FASTAL:5 INFO
000038 019 C
000039 019 C 1100 CONTINUE
000040 019 RSHORT(7) = UIN
000041 019 RSHORT(8) = RULE
000042 019 RSHORT(10) = TMPD
000043 019 RSHORT(11) = SFRB9
000044 019 RSHORT(12) = LRE
000045 019 RSHORT(13) = RBRNCH
000046 019 RSHORT(22) = BLANK
000047 019 RSHORT(23) = RSRC15
000048 019 RSHORT(24) = RSRC67
000049 019 RSHORT(25) = RSRCB9
000050 022 RSHORT(26) = ONE
000051 022 IF (RSRCA.EQ.FIVE) RSHORT(26) = B
000052 022 RSHORT(27) = RSRCB
000053 022 RSHORT(28) = RSRC
000054 019 RSHORT(29) = MCODE9
000055 019 RSHORT(33) = RTPSN
000056 019 RSHORT(37) = RUNT01
000057 019 RSHORT(38) = RUNT02
000058 019 RSHORT(39) = RUNT03
000059 019 RSHORT(40) = RUNT04
000060 019 RSHORT(41) = RSOFF
000061 019 RSHORT(42) = RSMOF
000062 019 RSHORT(43) = RSENL
000063 019 RSHORT(44) = RSAGK
000064 019 RLOCCO = NOTAB(9)
000065 019 RSTNM1 = NOTAB(15)
000066 019 RSTNM2 = NOTAB(16)
000067 019 RCOMPO = NOTAB(18)
000068 019 C
000069 019 C *** COMPUTE UNMBR ***
000070 019 C
000071 019 C IF (NEWSRC.NE.OLDSRC) GOTO 1200
000072 019 C THEN
000073 019 C UNMBR = UNMBR + 1
000074 019 C GOTO 1300
000075 019 C ELSE
000076 019 C 1200 UNMBR = 1
000077 019 C OLDSRC = NEWSRC
000078 019 C END-IF
000079 019 C 1300 CONTINUE
000080 019 RSHORT(35) = UNMBR
000081 019 C
000082 019 C *** LOCATE PECOD FROM PTAB AND TRANSFER IT ***
000083 019 C
000084 019 C I = 1
```



```
000085 019 PECOD = ASTER
000086 019 SRC12 = RSRC15 / 1000
000087 019 C
000088 019 C DOWHILE (I *LT. 24)
000089 019 C
000090 019 2000 IF (I *GT. 23) GOTO 2200
000091 019 C
000092 019 IF (SRC12 *NE. PTAB(1,1)) GOTO 2100
000093 019 C THEN
000094 019 C SRC12 MATCHES PTAB:5 SRC12
000095 019 C PECOD = PTAB(2,1)
000096 019 C GOTO 2200
000097 019 C ELSE
000098 019 2100 I = I + 1
000099 019 C END-IF
000100 019 C GOTO 2000
000101 019 C END-DO
000102 019 C
000103 019 2200 CONTINUE
000104 019 RSHORT(36) = PECOD
000105 019 C *** SET-UP AUTHORITY FIELDS ***
000106 019 C
000107 019 3000 CONTINUE
000108 019 C
000109 019 C *** COMPUTE UIC36 ***
000110 019 C
000111 019 UIC36 = NXUIC
000112 019 NXUIC = NXUIC + 1
000113 019 ENCODE(16,2,RSHORT(3)) NOTAB(19), UIC36
000114 019 RUIC = RSHORT(3)
000115 019 2 FORMAT (A2,J4)
000116 019 C
000117 019 C *** ENCODE INFO INTO RECORD ***
000118 019 C
000119 019 ENCODE (150,1,RECORD) RSHORT
000120 019 1 FORMAT (2A1,2A6,A1,A2,J3,A4,A1,J2,A3,A1,J4,3A3,A6,A3,A1,A3,
* J5,2A2,4A1,A5,A6,A3,J5,A1,J4,A6,A3,3A6,4A5,A1,A1,A5)
000121 020
000122 019 C
000123 019 C *** INCREMENT TIME-PERIOD COUNTER ***
000124 019 C
000125 019 C KNTROD(TMPD) = KNTROD(TMPD) + 1
000126 019 C
000127 019 RETURN
000128 019 C
END ELT.
```

WELT.L FSFM.MAIN

ELT007 RL1870 08/16-07:40:32-(18.)

```
000001 014 C
000002 014 C M A I N - R O U T I N E
000003 014 C
000004 014 C
000005 014 C
000006 014 C
```

RENE: R. PLOURDE
USACAA/MRD JUNE 1976


```

000007 014 IMPLICIT INTEGER (A-Z)
000008 014 COMMON /TABLES/ NOTAB(20), PTAB(2,23), RDOTAB(7,10),
000009 014 STPI(6,10), KNTRDD(10)
000010 014 COMMON /RECORD/ RECORD(25)
000011 014 COMMON /FASTAL/ RBRNCH,MSRC15,MSRC67,MSRC89,MSRCA,MSRCB,MSRCC,
000012 014 KTPSN,RUNTD1,RUNTD2,RUNTD3,RUNTD4,RSOFF,MSWOF,
000013 014 KSENL,RSAGR,WIN,ROLE,TPMD,RE,UNHBR,OLDSRC,NAUIC,
000014 014 RSTR,REWSEQ,KSTNM1,KSTNM2,KUIC,RLCCO,RCOMP,SEF89
000015 014 COMMON /MCAND/ MSRC15,MSRC67,MSRC89,MSRCA,MSRCB,MSRCC,COMP,
000016 014 MC,UIC1ST,DAMPL,UIC,MUNTD1,MUNTD2,MUNTD3,MUNTD4,
000017 014 MCSEQ,MTPSN,MBRNCH,MSAGR,LOCCO,MSTNM1,MSTNM2,ROBCO,SEF89
000018 014 INTEGER KEY(24), LIST(45), BUFF(24)
000019 014 DATA LINE/100/
000020 014 DATA KEY/5,7,12,0,0,1,12,3,6,0,0,2,25,1,36,0,0,3,
000021 014 1,13,36,0,0,4,9999,216,0/
000022 014 DATA NXUIC/1/, AUTH13/3H000/, AUTH47/4H0000/, AUTH8/1H0/,
000023 014 AUTH9A/2H99/, AUTHBD/3H999/, AUTHC/1H0/, BLANK/1H /
000024 014 DATA PFILE/1/, RDFILE/12/, NOFILE/13/
000025 014 DATA FMFILE/1/, FRFILE/2/, OVFILE/3/, MRFILE/8/
000026 014 1 FORMAT (1X,J5,A2,3A1,A5,A6,A2,3A1,A2,A3,3A6,J5,A3,A6,2A3,A5,
000027 014 A1,A2,A3,16,A6)
000028 014 2 FORMAT(1X,15,1X,A2,1X,12,1X,A2,3(1X,A1),1X,A3,3(1X,A6),1X,A4,
000029 014 3(1X,A1),1X,J5,4(1X,A5),1X,13,1X,A2,1X,13,16,A3)
000030 014 5 FORMAT(15,A6,A3,A4,A1,A2,A3,A1)
000031 014 C
000032 014 CALL SUPEN3 ($100,$5000,25,25,KEY)
000033 014 C
000034 014 100 READ (PFILE,11) PTAB
000035 014 11 FORMAT (12,2X,A6)
000036 014 C
000037 014 READ (RDFILE,12) RDOTAB
000038 014 12 FORMAT (714)
000039 014 C
000040 014 READ (NOFILE,13) NOTAB
000041 014 13 FORMAT (A6)
000042 014 C
000043 014 C
000044 014 FLAG = 1
000045 014 C
000046 014 C DOWHILE (NOT EOF)
000047 014 C
000048 014 1000 IF (FLAG.NE.1 .AND. FLAG.NE.2) GOTO 1U20
000049 014 C THEN
000050 014 C GET NEXT MATCH=CANDIDATE=RECORD
000051 014 READ (FMFILE,1,END=2000) MSRC15,MSRC89,COMP,MC,UIC1ST,
000052 014 DAMPL,UIC,MSRC67,MSRCA,MSRCB,MSRCC,
000053 014 MBRNCH,MUNTD1,MUNTD2,MUNTD3,MUNTD4,
000054 014 MTPSN,ROBCO,MSTNM1,MSTNM2,LOCCO,
000055 014 MSAGR,MADCO1,MADCO2,MADCO3,MCSEQ,SEF89
000056 014 N89 = FLU(0,12,MSRC89)
000057 014 MKEY = MSRC15 * 5000 + N89
000058 014 C END-IF
000059 014 C
000060 014 1U20 IF (FLAG.NE.1 .AND. FLAG.NE.3) GOTO 1U40
000061 014 C THEN
000062 014 C GET NEXT REQUIREMENT=RECORD
000063 014 READ (FRFILE,2,END=3000) RSRC15,MSRC89,TPMD,MSRC67,MSRCB,

```

```
000064 014 *
000065 014 *
000066 014 *
000067 014 *
000068 014 N89 = FL010,12,RSRC89)
000069 014 RKEY = RSRC15 * 5000 + N89
000070 014 C END=IF
000071 014 C
000072 014 C *** CHECK FOR SHORT-FALL (RKEY ) MKEY) ***
000073 014 C 1040 IF (RKEY .GE. MKEY) GOTO 1060
000074 014 C THEN
000075 014 C WE HAVE A SHORT-FALL CREATE A NOTIONAL UNIT
000076 014 C FLAG = 3
000077 014 C TSHORT = TSHORT + 1
000078 014 C CALL SHORT (RKEY)
000079 014 C CALL SRREL (RECORD,25)
000080 014 C CALL PRT (LINE,1)
000081 014 C GOTO 1200
000082 014 C ELSE
000083 014 C *** CHECK FOR OVERAGE (RKEY .GT. MKEY)
000084 014 C 1060 IF (RKEY .EG. MKEY) GOTO 1080
000085 014 C THEN
000086 014 C WE HAVE AN OVERAGE
000087 014 C FLAG = 2
000088 014 C TOVER = TOVER + 1
000089 014 C CALL OVER
000090 014 C CALL SRREL (RECORD,25)
000091 014 C CALL PRT (LINE,3)
000092 014 C GOTO 1180
000093 014 C ELSE
000094 014 C *** CHECK TIME-PERIOD ***
000095 014 C 1080 IF (TMPD .EG. 1) GOTO 1100
000096 014 C THEN
000097 014 C WE HAVE A GOOD MATCH
000098 014 C FLAG = 1
000099 014 C TMATCH = TMATCH + 1
000100 014 C CALL MATCH
000101 014 C CALL SRREL (RECORD,25)
000102 014 C CALL PRT (LINE,2)
000103 014 C GOTO 1160
000104 014 C ELSE
000105 014 C *** CHECK MCODE FOR U OR I ***
000106 014 C 1100 IF (MC.NE.U .AND. MC.NE.I) GOTO 1120
000107 014 C THEN
000108 014 C WE HAVE A GOOD MATCH
000109 014 C FLAG = 1
000110 014 C TMATCH = TMATCH + 1
000111 014 C CALL MATCH
000112 014 C CALL SRREL (RECORD,25)
000113 014 C CALL PRT (LINE,2)
000114 014 C GOTO 1140
000115 014 C ELSE
000116 014 C WE HAVE AN IN-COUNTRY SHORT-FALL
000117 014 C FLAG = 3
000118 014 C TSHORT = TSHORT + 1
000119 014 C CALL SHORT (RKEY)
000120 014 C CALL SRREL (RECORD,25)
```

●●UNCLASSIFIED●●

FMA PROGRAM MATCH

Line	Code	Statement
014	000121	C
014	000122	C
014	000123	1140
014	000124	C
014	000125	1160
014	000126	C
014	000127	1180
014	000128	C
014	000129	1200
014	000130	C
014	000131	END=00
014	000132	C
014	000133	C
014	000134	C
014	000135	C
014	000136	C
014	000137	C
014	000138	C
014	000139	C
014	000140	C
014	000141	C
014	000142	C
014	000143	C
014	000144	C
014	000145	C
014	000146	C
014	000147	C
014	000148	C
014	000149	C
014	000150	C
014	000151	C
014	000152	C
014	000153	C
014	000154	C
014	000155	C
014	000156	C
014	000157	C
014	000158	C
014	000159	C
014	000160	C
014	000161	C
014	000162	C
014	000163	C
014	000164	C
014	000165	C
014	000166	C
014	000167	C
014	000168	C
014	000169	C
014	000170	C
014	000171	C
014	000172	C
014	000173	C
014	000174	C
014	000175	C
014	000176	C
014	000177	C

FMA PROGRAM MATCH

●●UNCLASSIFIED●●


```
000178 014 CALL SRREL (RECORD,25)
000179 014 CALL PRT (LINE,3)
000180 014 GOTO 3200
000181 014 C END=DO
000182 014 C
000183 014 C *** TIME TO SORT THE MATCH=RESULTS ***
000184 014 C
000185 014 C 4000 CALL SSORT
000186 014 C
000187 014 C *** COMPUTE RDU ***
000188 014 C
000189 014 C 5000 TP = I
000190 014 WRITE(6,77) TMATCH,TSHORT,TOVER
000191 014 77 FORMAT(' THATCH=',16,' TSHORT=',16,' TOVER=',16)
000192 014 C
000193 014 C DOWHILE (TP .LT. 10)
000194 014 C
000195 014 C 5100 IF (TP .GT. 10) GOTO 5300
000196 014 C
000197 014 C IF (KNTRDD(TP) .EQ. 0) GOTO 5200
000198 014 C THEN
000199 014 SP = I
000200 014 WRITE(6,66) TP,KNTRDD(TP)
000201 014 66 FORMAT(' TP=',12,' KNTRDD(TP)=',16)
000202 014 AV = KNTRDD(TP) / 6
000203 014 K = KNTRDD(TP) - AV * 6
000204 014 C
000205 014 C DOWHILE (SP .LT. 7)
000206 014 C
000207 014 C 5120 IF (SP .GT. 6) GOTO 5200
000208 014 C
000209 014 C IF (SP .GT. R) GOTO 5140
000210 014 C THEN
000211 014 L = AV + 1
000212 014 GOTO 5160
000213 014 C ELSE
000214 014 L = AV
000215 014 C END-IF
000216 014 C CONTINUE
000217 014 C M = I
000218 014 C
000219 014 C DOWHILE (M .LT. L)
000220 014 C
000221 014 C 5180 IF (M .GT. L) GOTO 5199
000222 014 C CALL SKRET (RECORD,DUMMY,$9999)
000223 018 DECODE (150,3,RECORD) LIST
000224 014 TR = TR + 1
000225 015 IF (LIST(45) .NE. '0') GOTO 5182
000226 016 TNOT = TNOT + 1
000227 015 GOTO 5188
000228 015 5182 IF (LIST(45) .NE. '1') GOTO 5184
000229 015 TMTCH = TMTCH + 1
000230 015 GOTO 5188
000231 015 5184 IF (LIST(45) .NE. '2') GOTO 5186
000232 015 750V = 750V + 1
000233 015 GOTO 5188
000234 015 5186 WRITE (6,89) LIST(45)
```


UNCLASSIFIED

```
000235 015 5188 CONTINUE
000236 017 89 FORMAT (' *** LIST(45) =',A6,' ***')
000237 014 LIST(25) = LIST(11)
000238 014 ENCODE (3,88,LIST(11)) MDUTAB(SP,TP)
000239 014 88 FORMAT (J3)
000240 014 ENCODE (150,3,BUFF) LIST
000241 014 WRITE (6,1111) (BUFF(10),10=1,12)
000242 014 WRITE (6,1111) (BUFF(JD),JD=13,24)
000243 014 1111 FORMAT (20A6)
000244 014 WRITE(MRFILE) BUFF
000245 014 3 FORMAT (2A1,2A6,A1,A2,A3,A4,A1,A2,A3,A1,3A2,3A3,A6,A3,A1,A3,
000246 014 * J5,2A2,4A1,A5,A6,A3,J5,A1,A4,A6,A3,3A6,4A5,A1)
000247 014 4 FORMAT (2A1,2A6,A1,A2,A3,A4,A1,A2,A3,A1,A5,A2,3A3,A6,A3,A1,A5,
000248 014 * 2A2,4A1,2A6,A3,15,A1,14,A6,A3,3A6,4A5,A1)
000249 014 IF (LIST(45) .NE. '2') M = M + 1
000250 014 GOTO 5180
000251 014 C END=DO
000252 014 C
000253 014 5199 CONTINUE
000254 014 WRITE(6,39) TP,SP,M,L,AV,R,TR
000255 014 39 FORMAT(' TP',13,' SP',12,' M',16,' L',16,' AV',16,
000256 014 * R',16,' TR',16)
000257 014 SP = SP + 1
000258 014 GOTO 5120
000259 014 C END=DO
000260 014 C
000261 014 5200 CONTINUE
000262 014 67 FORMAT (' TP',12,' TR',16)
000263 014 TP = TP + 1
000264 014 GOTO 5100
000265 014 C END=DO
000266 014 C
000267 014 9999 CONTINUE
000268 014 IF (TR.EQ. TMATCH*SHORT+TOVER) GOTO 5300
000269 014 WRITE (6,99)
000270 014 99 FORMAT ('1 *** ERROR OCCURRED WHILE COMPUTING RDD ***')
000271 014 5300 CONTINUE
000272 014 WRITE (6,55) TR,TSOV,TNOT,TMTC
000273 014 55 FORMAT(' TOTAL NUMBER OF RECORDS RETURNED FROM SORT =',16,316)
000274 014 ENDFILE OVFILE
000275 014 ENDFILE MRFILE
000276 014 STATUS = CSF(2,'ERRKPT 19 ')
000277 014 STOP
000278 014 END

END ELT.
```

```
9ELT.L F5FM-MATCH
ELT007 RL1870 08/16-07:40:34-(1,)
000001 000 C
000002 000 C S U B M A T C H
000003 000 C
000004 000 C SUBROUTINE MATCH
000005 000 C
000006 000 C IMPLICIT INTEGER (A-Z)
```

UNCLASSIFIED

FNA PROGRAM MATCH

●●UNCLASSIFIED●●

```

000007 000 COMMON /TABLES/ NOTAB(20), PTAB(2,23), RDOOTAB(7,10),
000008 000 STP(6,10), KNTRDD(10)
000009 000 COMMON /REGREC/ RECORD(25)
000010 000 COMMON /FASTAL/ KBRNCH,MSRCL5,MSRCL6,MSRCL7,MSRCL8,MSRCL9,MSRCA,MSRCB,MSRCC,
000011 000 HTPSN,RUNTD1,RUNTD2,RUNTD3,RUNTD4,RSOFF,RSWOF,
000012 000 KSENL,RSAGR,UIN,RULE,TPMD,LMR,UMBR,OLDSRC,MAUIC,
000013 001 HSTR,RCSEQ,STNM1,STNM2,RUIC,PLOCCO,RCOMPO,SFRB9
000014 000 COMMON /MCAND/ MSRCL5,MSRCL6,MSRCL8,MSRCL9,MSRCA,MSRCB,MSRCC,COMPO,
000015 000 MC,UICIST,DAMPL,UIC,MUNTD1,MUNTD2,MUNTD3,MUNTD4,
000016 001 MCSEQ,HTPSN,KBRNCH,MSAGK,LUCCO,MSNM1,STNM2,ROBCO,SFMB9
000017 000 INTEGER RMATCH(47)
000018 000 DATA RMATCH/44*6H
000019 000 DATA ONE/1H1/
000020 000 C
000021 000 C *** SET-UP THE AUTHORITY FIELDS IN RMATCH ***
000022 000 C
000023 000 RMATCH(2) = COMPU
000024 000 RMATCH(3) = UIC
000025 000 RMATCH(7) = UIN
000026 000 RMATCH(8) = RULE
000027 000 RMATCH(9) = ONE
000028 000 RMATCH(10) = TPMD
000029 001 RMATCH(11) = SFMB9
000030 000 RMATCH(12) = LRE
000031 000 RMATCH(22) = ROBCO
000032 000 RMATCH(29) = MC
000033 000 RMATCH(46) = UICIST
000034 000 RMATCH(47) = DAMPL
000035 000 C
000036 000 C *** ENCODE INFO INTO RECORD ***
000037 000 C
000038 000 ENCODE (150,1,RECORD) RMATCH
000039 000 C
000040 000 C *** INCREMENT TIME-PERIOD COUNTER ***
000041 000 C
000042 000 KNTRDD(TPMD) = KNTRDD(TPMD) + 1
000043 000 C
000044 000 RETURN
000045 000 I FORMAT (2A1,2A6,A1,A2,J3,A4,A1,J2,A3,A1,JAZ,JAZ,A6,A3,A1,A3,
000046 000 A5,2A2,A1,A5,A6,A3,A5,A1,A4,A6,A3,3A6,A5,A1,A1,A5)
000047 000 END
000048 000

```

08ELT.L F5FM.OVER
ELT007 RL1870 08/16-07:40:35-(2.)

	C	O	V	E	R	A	G	E
000001	000							
000002	000							
000003	000							
000004	000							
000005	000							
000006	000							
000007	000							
000008	000							
000009	000							

```

000010 000 COMMON /REGR/ RECORD(25)
000011 000 COMMON /FASTAL/ RBNCH,RSNC15,RSNC67,RSNC89,RSRCA,RSRCB,RSRCC,
000012 000 RPSN,RUNT01,RUNT02,RUNT03,RUNT04,R5OFF,R5WOF,
000013 000 R5ENL,RSAGR,UIN,KULE,THPD,LRE,UNMR,OLD5RC,NAUIC,
000014 001 RSTR,RESEQ,RSTN1,MSTNM2,KUIC,LOCLO,COMPO,SFMB9
000015 000 COMMON /MCAND/ MSRC15,MSRC67,MSRC89,MSRCA,MSRCB,MSRCC,COMPO,
000016 000 MC,UIC1ST,DAMPL,UIC,MUNT01,MUNT02,MUNT03,MUNT04,
000017 001 MCSEQ,MTPSN,MRNCH,MSAGR,LOCLO,MSTNM1,MSTNM2,ROBCO,SFMB9
000018 000 C
000019 000 INTEGER OVLIST(47)
000020 000 DATA MCODEB/1H8/
000021 000 C
000022 000 DATA OVLIST/9*6H0000000,2*6H9999999,33*6H0000000,1H2,2*1H /
000023 000 OVLIST(22) = ROBCO
000024 000 OVLIST(10) = 9
000025 000 OVLIST(29) = MCODE8
000026 000 OVLIST(46) = UIC1ST
000027 000 OVLIST(47) = DAMPL
000028 000 OVLIST(2) = COMPO
000029 000 OVLIST(3) = UIC
000030 000 KNRDD(9) = KNRDD(9) + 1
000031 000 C
000032 000 ENCODE (150,1,RECORD) OVLIST
000033 000 C
000034 000 C D I T O
000035 002 1 FORMAT (2A1,2A6,A1,A2,A3,A4,A1,J2,A3,A1,3A2,3A3,A6,A3,A1,A3,
000036 000 AS,2A2,4A1,A5,A6,A3,A5,A1,A4,A6,A3,3A6,4A5,A1,A1,A5)
000037 000 C RETURN
000038 000 RETURN
000039 000 END

```

END ELT.

SHDG FMA PROGRAM ATL

••UNCLASSIFIED•• L,U

```
WELT.L FSATL.FIND
ELT007 RL1870 08/16-07:40:37-10.)
000001 000 C
000002 000 SUBROUTINE FIND (KEY,I,SIZE)
000003 000 IMPLICIT INTEGER (A = Z)
000004 000 COMMON /TTAB/ TABLE(5,100)
000005 000 C
000006 000 MAX = SIZE
000007 000 MIN = 1
000008 000 I = 0
000009 000 C
000010 000 C DOWHILE (I.LE.0 .AND. MIN.LE.MAX)
000011 000 C
000012 000 C 1000 IF (I.NE.0 .OR. MIN.GT.MAX) RETURN
000013 000 C
000014 000 MID = (MAX + MIN)/2
000015 000 IF (KEY.NE.TABLE(I,MID)) GOTO 1100
000016 000 C WE HAVE FOUND THE KEY
000017 000 I = MID
000018 000 TABLE(5,I) = 1
000019 000 GOTO 1200
000020 000 C ELSE
000021 000 C 1100 IF (KEY.LT.TABLE(I,MID)) GOTO 1120
000022 000 C THEN
000023 000 C POSSIBLE MATCH IS IN UPPER HALF OF TABLE
000024 000 MIN = MID + 1
000025 000 GOTO 1140
000026 000 C ELSE
000027 000 C POSSIBLE MATCH IS IN LOWER HALF OF TABLE
000028 000 1120 MAX = MID - 1
000029 000 C END-IF
000030 000 C CONTINUE
000031 000 C END-IF
000032 000 C 1200 CONTINUE
000033 000 C GOTO 1000
000034 000 C END-DO
000035 000 C
000036 000 C
END ELT.
```

```
WELT.L FSATL.GETTP
ELT007 RL1870 08/16-07:40:37-10.)
000001 000 C
000002 000 SUBROUTINE GETTP (RDD,TP)
000003 000 C
000004 000 IMPLICIT INTEGER (A = Z)
000005 000 COMMON /RTAB/ RDTAB(7,10)
000006 000 C
000007 000 I = 1
000008 000 C DOWHILE (I.LE.10)
000009 000 C
000010 000 C 1000 IF (I.GT.10) GOTO 2000
000011 000 C
000012 000 C
```



```
000013 000 IF (RDU.GT. RDTAB(7,1)) GOTO 1100
000014 000 THEN
000015 000 C
000016 000 C WE HAVE FOUND THE CORRECT TIME-PERIOD (TP)
000017 000 TP = 1
000018 000 RETURN
000019 000 ELSE
000020 000 1100 I = 1 + 1
000021 000 C END-IF
000022 000 GOTO 1000
000023 000 C END-DO
000024 000 C
000025 000 2000 CONTINUE
000026 000 WRITE (6,1) RDD
000027 000 I FORMAT (10) *** RDD = '13.' WAS NOT FOUND IN RDTAB ***
000028 000 TP = 0
000029 000 RETURN
000030 000 END
```

END ELT.

```
9ELT.L FSATL*MAIN
ELT007 RL1870 08/16-07:40:39-(13.)
000001 000 C
000002 000 C A B O V E - T H E - L I N E
000003 000 C
000004 000 C
000005 000 C
000006 000 C
000007 000 IMPLICIT INTEGER (A-Z)
000008 000 COMMON /TTAB/ TABT(5,100)
000009 000 COMMON /RTAB/ RDTAB(7,10)
000010 000 INTEGER LIST(27)
000011 000 EQUIVALENCE (LIST(1),SRC15), (LIST(2),SRC89), (LIST(3),CUMPO),
000012 000 (LIST(4), MC), (LIST(5),UIC1ST), (LIST(6),DAMPL),
000013 000 (LIST(7), UIC), (LIST(8),SRC67), (LIST(9), SRC4),
000014 000 (LIST(10), SRCB), (LIST(11), SRCC), (LIST(12),BRNCH),
000015 000 (LIST(13),UNTD1), (LIST(14),UNTD2), (LIST(15),UNTD3),
000016 000 (LIST(16),UNTD4), (LIST(17),TPSN ), (LIST(18),RORCO),
000017 000 (LIST(19),STNM1), (LIST(20),STNM2), (LIST(21),LUCCO),
000018 000 (LIST(22), SAGR), (LIST(23),ADCO1), (LIST(24),ADCO2),
000019 000 (LIST(25),ADCO3), (LIST(26), SEQ), (LIST(27),MPFLAG)
000020 000 DATA RDDFIL(1), TPSN/12, ATFL/13, FILATL/14/
000021 000 DATA YES/3YES/, NO/2HNO/
000022 000 DATA ZERO/1H0/, ONE/1H1/, BLANK/6H /, ASTER/1H*/
000023 000 C
000024 000 C
```

RENE, K. PLOURDE
USACCA/MRD JUNE 1976

```
000025 000 C *** GET RDD TABLE FROM RDDFIL FILE ***
000026 000 C
000027 000 I = 1
000028 000 C
000029 000 C DO WHILE (I.LE. 10)
000030 000 C
000031 000 1000 IF (I.GT. 10) GOTO 1999
000032 000 C
000033 000 READ (RDDFIL,1) (RDTAB(J,1),J=1,7)
```

```
000034 000 WRITE (6,2) I, (RDTAB(J,1),J=1,7)
000035 000 I = I + 1
000036 000 GOTO 1000
000037 000 C END=DO
000038 000 C
000039 000 1999 CONTINUE
000040 000 I = 1
000041 000 C
000042 000 C *** LOAD TPSN TABLE FROM TPSNF FILE TO TABT TABLE ***
000043 000 C
000044 000 C DOWHILE (NOT EOF FOR TPSNF)
000045 000 C
000046 000 2000 READ (TPSNF,3,END=2999) (TABT(J,1),J=1,3)
000047 000 CALL GETTP (TABT(3,1),TABT(4,1))
000048 000 WRITE (6,4) I, (TABT(J,1),J=1,4)
000049 000 I = I + 1
000050 000 GOTO 2000
000051 000 C END=DO
000052 000 C
000053 000 2999 CONTINUE
000054 000 TSIZE = 1 - 1
000055 000 C
000056 000 C *** PROCESS ATL RECORDS ***
000057 000 C
000058 000 C DOWHILE (NOT EOF FOR ATLFIL)
000059 000 C
000060 000 3000 READ (ATLFIL,5,END=3999) LIST, FLGATL
000061 000 CALL FIND (TPSN,INDEX,TSIZE)
000062 000 IF (INDEX.EQ. 0) GOTO 3300
000063 000 C THEN
000064 000 C *** ADC01 = 1 ***
000065 000 ADC01 = TABT(2,INDEX)
000066 000 LOFLAG = BLANK
000067 000 IF (TABT(2,INDEX) .NE. ONE) GOTO 3100
000068 000 C THEN
000069 000 C *** HPFLAG ***
000070 000 IF (HPFLAG.EQ. YES .AND. ADC03.NE. BLANK) GOTO 3020
000071 000 C THEN
000072 000 ADC02 = TABT (4,INDEX)
000073 000 ADC03 = TABT(3,INDEX)
000074 000 GOTO 3040
000075 000 C ELSE
000076 000 C COMPUTE NEW TP BASED ON INPUT ADC03
000077 000 3020 DECODE (3,7,ADC03) ADC03
000078 000 CALL GETTP (ADC03,ADC02)
000079 000 C END-IF
000080 000 3040 CONTINUE
000081 000 GOTO 3200
000082 000 C ELSE
000083 000 3100 ADC02 = 0
000084 000 IF (HPFLAG.EQ. YES .AND. ADC03.NE. BLANK) GOTO 3120
000085 000 C THEN
000086 000 ADC03 = TABT(3,INDEX)
000087 000 GO TO 3200
000088 000 C END-IF
000089 000 3120 DECODE (3,7,ADC03) ADC03
000090 000 C END-IF
```

```

000091 000 3200 CONTINUE
000092 000 GOTO 3400
000093 000 C ELSE
000094 000 3300 LOFLAG = ASTER
000095 000 C END-IF
000096 000 3400 CONTINUE
000097 000 LIST(27) = LOFLAG
000098 003 WRITE (FILATL,8) LIST, FLGATL
000099 000 GOTO 3000
000100 000 C END-DO
000101 000 C
000102 000 3999 CONTINUE
000103 000 I = 1
000104 000 C DO-WHILE ( I .LE. TSIZE)
000105 000 C
000106 000 C
000107 000 4000 IF ( I .GT. TSIZE) GOTO 4999
000108 000 C
000109 000 IF (TABT(5,1) .NE. 0) GOTO 4100
000110 000 C THEN
000111 000 C PRINT THAT THIS TPSN WAS NOT USED
000112 000 WRITE (6,6) I, (TABT(J,1),J=1,5)
000113 000 C END-IF
000114 000 4100 CONTINUE
000115 000 I = I + 1
000116 000 GOTO 4000
000117 000 C END-DO
000118 000 C
000119 000 4999 CONTINUE
000120 000 C
000121 000 WRITE (6,101)
000122 000 101 FORMAT(' *** ABOVE-THE-LINE FINI ***')
000123 000 STATUS = CSF (2,'WBKPT 11 ')
000124 000 STATUS = CSF (2,'WBKPT 12 ')
000125 000 STATUS = CSF (2,'WBKPT 13 ')
000126 000 STATUS = CSF (2,'WBKPT 14 ')
000127 000 STOP
000128 000 1 FORMAT (714)
000129 000 2 FORMAT (1X,13,716)
000130 000 4 FORMAT (1X,13,2X,J5,3X,A1,3X,J3,2X,J2)
000131 000 3 FORMAT (J5,2X,A1,2X,J3)
000132 000 5 FORMAT (1X,J5,A2,3A1,A5,A6,A2,3A1,A2,A3,3A6,J5,A3,A6,2A3,A5,
000133 003 A1,J2,A3,16,A6,A3)
000134 000 6 FORMAT (1H0,13,1X,*** TPSN='J5,' WAS NOT USED ***',2X,A1,
000135 000 J3,2X,J2,2X,12)
000136 000 7 FORMAT (J3)
000137 003 8 FORMAT (1X,J5,A2,3A1,A5,A6,A2,3A1,A2,A3,3A6,J5,A3,A6,2A3,A5,
000138 003 A1,J2,J3,16,A6,A3)
000139 000 END

```

END ELT.

••UNCLASSIFIED•• L,U

BHOG FMA PROGRAM ATL


```

FMA PROGRAM ALT
BELT.L FSALT:MAIN
ELT007 RL1870 08/16-07:40:40-(1,1)
000001 000 C
000002 000 C
000003 000 C
000004 000 C
000005 000 C
000006 000 C
000007 000 C
000008 000 C
000009 000 C
000010 000 C
000011 000 C
000012 000 C
000013 000 C
000014 000 C
000015 000 C
000016 000 C
000017 000 C
000018 000 C
000019 000 C
000020 000 C
000021 000 C
000022 000 C
000023 000 C
000024 000 C
000025 000 C
000026 000 C
000027 000 C
000028 000 C
000029 000 C
000030 000 C
000031 000 C
000032 000 C
000033 000 C
000034 000 C
000035 000 C
000036 000 C
000037 000 C
000038 000 C
000039 000 C
000040 000 C
000041 000 C
000042 000 C
000043 001 C
000044 000 C
000045 000 C
000046 000 C
000047 000 C
000048 000 C
000049 000 C
000050 000 C
000051 000 C
000052 000 C
000053 000 C
000054 000 C
000055 000 C

A L T E R N A T E - T H E A T E R

RENEG K. PLOURDE
USACCA/MRD JUNE 1976

IMPLICIT INTEGER (A = 2)
INTEGER ALTAB(6,100), ROD(100), LIST(26)
DATA ALTFIL/1/, ALTFIL/12/, FILALT/13/

EQUIVALENCE (LIST(1),SRC15), (LIST(2),SRC89), (LIST(3),COMPO),
              (LIST(4), MC), (LIST(5),UIC1ST), (LIST(6),DAMPL),
              (LIST(7), UIC), (LIST(8),SRC67), (LIST(9), SRCA),
              (LIST(10), SRC8), (LIST(11), SRCC), (LIST(12),BRNCH),
              (LIST(13),UNTD1), (LIST(14),UNTD2), (LIST(15),UNTD3),
              (LIST(16),UNTD4), (LIST(17),TPSN ), (LIST(18),MURCO),
              (LIST(19),STNM1), (LIST(20),STNM2), (LIST(21),LOCCO),
              (LIST(22), SAGR), (LIST(23),ADCO1), (LIST(24),ADCO2),
              (LIST(25),ADCO3), (LIST(26), SEQ)

WRITE (6,21)
21 FORMAT ('ADCO1 LOWER-LIMIT UPPER-LIMIT INCRE NEW=ADCO1',//)
11 FORMAT ('X,A1,8X,J3,10X,J3,7X,J3,7X,A1)
1 = 1

C DOWNHILE (NOT EOF FOR ALTFIL)
1000 READ (ALTFIL,1,END=1999) (ALTAB(J,1),J=1,5)
1 FORMAT (A1,2X,J3,2X,J3,2X,J3,2X,A1)
WRITE (6,11) (ALTAB(J,1),J=1,5)
ROD(1) = ALTAB(2,1) - ALTAB(4,1)
1 = 1 + 1
GOTO 1000
C END=DO

1999 CONTINUE
TSIZE = 1 - 1

C DOWNHILE (NOT EOF FOR ALTFIL)
2000 READ (ALTFIL,8,END=2999) LIST
8 FORMAT ('X,J5,A2,3A1,A5,A6,A2,3A1,A2,A3,3A6,J5,A3,A6,2A3,A5,
          A1,J2,J3,16,A6)
1 = 1
J = 0

C DOWNHILE (1,LE,TSIZE,AND,J.EQ,0)
2100 IF (1,GT,TSIZE,OR,J,NE,0) GOTO 2199
IF (ADCO1,NE,ALTAB(1,1)) GOTO 2120
THEN
ADCO1'S MATCH
J = 1
ROD(J) = ROD(J) + ALTAB(4,J)

```


UNCLASSIFIED

FMA PROGRAM ALT

```
000056 000 IF (RDD(J) .GT. ALTAB(5,J)) RDD(J) = ALTAB(2,J)
000057 000 ADC01 = ALTAB(5,J)
000058 000 IF (ADC03 .EQ. 999) ADC03 = RDD(J)
000059 000 ADC02 = 0
000060 000 ALTAB(6,J) = 1
000061 000 GOTO 2140
000062 000 ELSE
000063 000 I = 1
000064 000 END-IF
000065 000 CONTINUE
000066 000 GOTO 2100
000067 000 C END-DO
000068 000 C
000069 000 C 2199 CONTINUE
000070 000 C
000071 000 IF (J .NE. 0) GOTO 2200
000072 000 THEN
000073 000 C ADC01 WAS NOT FOUND
000074 000 WRITE (6,2) ADC01
000075 000 2 FORMAT (1H0,300 ADC01 = ,A1, WAS NOT FOUND IN THE TABLE 000)
000076 000 LFLAG = 1
000077 000 GOTO 2220
000078 000 ELSE
000079 000 C 2200 WRITE (FILALT,8) LIST
000080 000 C ENDIF
000081 000 C 2220 CONTINUE
000082 000 GOTO 2000
000083 000 C END-DO
000084 000 C
000085 000 C 2999 CONTINUE
000086 000 WRITE (6,101)
000087 000 101 FORMAT ('000 ALTERNATE-THEATER FINI 000')
000088 000 C
000089 000 STATUS = CSF (2,'BBRKPT 11 ')
000090 000 STATUS = CSF (2,'BBRKPT 12 ')
000091 000 STATUS = CSF (2,'BBRKPT 13 ')
000092 000 STOP
000093 000 END
```

END ELT.

UNCLASSIFIED .L.U

ENDG FMA PROGRAM LAYIN

UNCLASSIFIED

FMA PROGRAM ALT

WELT.L FSLAY:MAIN
ELT007 RL1870 06/16-07:40:42-(10.)

000001 000 C
000002 000 C
000003 000 C
000004 000 C
000005 000 C
000006 000 C
000007 000 C
000008 000
000009 000
000010 000
000011 000
000012 000 C
000013 000
000014 000
000015 000
000016 000
000017 000
000018 000
000019 000
000020 000
000021 000
000022 000
000023 000
000024 000
000025 000
000026 000
000027 000
000028 000
000029 000
000030 000
000031 000
000032 000
000033 000
000034 000
000035 000
000036 000 C
000037 000
000038 000
000039 000
000040 000
000041 000
000042 000
000043 000
000044 000
000045 000 C
000046 000
000047 000 C
000048 000 C
000049 000 C
000050 000 C
000051 000
000052 000 C
000053 000 C
000054 000 C
000055 000 C

RENEW R. PLOURDE
USACAA/MRD JUNE 1976

IMPLICIT INTEGER (A = 4)
COMMON /RTAB/ RTOTAB(7,10)
INTEGER KEY(241), NOTREC(52), FASREC(52)
INTEGER ALTAB(5), ATLTAB(8), LOTAB(6), HPTAB(6), MRTAB(45)
INTEGER NINDEX(44), MRREC(24)

INTEGER FASTAB(99), NOTTAB(99)
DATA KEY/1,7,42,0,0,1,99999,234*0/
DATA FAS/3HFAS/, NOT/3HNOT/, ZERO/6H000000/, ONE/6H111111/
DATA COMPO/5HCOMPO/, X/1HX/, NINES/6H999999/
DATA ALTFILE/11/, ATLFIL/12/, LUFIL/13/, MPFILE/14/
DATA MRFILE/15/, FASFIL/16/, FILEF/17/, RDPFIL/10/
DATA (NOTTAB(J),J=1,75)/75*6H /
DATA (NOTTAB(98)/6H /, NOTTAB(99)/6H /
DATA (NOTTAB(K),K=83,97)/15*6H000000/
DATA NEXT/3HFAS/, ALTUIC/-1/, ATLUIC/-1/, LOUIC/-1/, HPUIC/-1/
DATA MUIC/-1/, BLANK/1H /
DATA NINDEX/1,2,3,4,5,8,9,10,11,12,13,14,15,16,24,31,
33,36,38,39,50,53,55,56,57,58,59,60,61,62,64,65,
70,72,76,77,78,79,80,81,93,94,95,96/
\$
\$
EQUIVALENCE (ALTAB(1),ALTUIC), (ATLTAB(2),ATLUIC)
EQUIVALENCE (LOTAB(1),LOUIC), (HPTAB(2),HPUIC)
EQUIVALENCE (MRTAB(3),MUIC), (FASTAB(3),FASUIC)
EQUIVALENCE (ATLTAB(7),ATLFLG), (LOTAB(6),LOTYPE)
EQUIVALENCE (MRTAB(45),MRFLAG), (ATLTAB(8),FLGALT)
EQUIVALENCE (FASTAB(9),AUTH13), (FASTAB(10),AUTH47)
EQUIVALENCE (FASTAB(11),AUTH8), (FASTAB(12),AUTH9A)
EQUIVALENCE (FASTAB(13),AUTHBD), (FASTAB(14),AUTH)
EQUIVALENCE (FASTAB(53),ROBCO)

1 FORMAT (1X,2A1,2A6,A1,A6,A4,A2,A3,A4,A1,A2,A3,3A2,A6,2A2,
A5,A3,A2,A3,2A2,5A1,A3,A5,2A3,A2,A3,A2,A6,A3,A2,
A1,A2,A1,A3,A1,A2,A3,5A1,A3,A1,A5,2A2,4A1,A5,A2,
A6,A3,A5,A6,A5,A1,A5,4A1,A5,A4,A6,A3,3A6,A3,15A5,
A4,A1)
3 FORMAT (2A1,2A6,A1,A2,A3,A4,A1,A2,A3,A1,3A2,3A3,A6,A3,
A1,A3,A5,2A2,4A1,A5,A6,A3,A5,A1,A4,A6,A3,3A6,
4A5,A1)
2 FORMAT (16X,A6,33X,A3,17X,A1,A2,A3,6X,A6)

CALL SOPEN3 (\$100,\$2000,24,4,KEY)

100 CONTINUE

*** GET RDD TABLE FROM RDDFIL FILE ***


```
000113 000 COUNT = COUNT + 1
000114 000 C
000115 000 C *** COMPARE ALT-TH (UIC) WITH FAS (UIC) ***
000116 000 C
000117 000 IF (ALTUIC *NE* FASUIC) GOTO 3400
000118 000 C THEN
000119 000 C PROCESS ONE EACH ALT-TH WITH THE CURRENT FAS-REC
000120 000 TALT = TALT + 1
000121 000 ROBCO = ALTTAB(2)
000122 000 AUTH8 = FASTAB(45)
000123 000 AUTH9A = BLANK
000124 000 AUTHBD = ALTTAB(5)
000125 000 ENCODE (2,102,AUTH1) BLANK, ALTTAB(3)
000126 000 AUTH13 = BLANK
000127 000 AUTH47 = BLANK
000128 000 NEXT = FAS
000129 000 IF (MOD(COUNT,50) *EQ* 0) CALL PRTFAS (COUNT,1,FASTAB)
000130 000 ENCODE (312,1,FASREC) FASTAB
000131 000 WRITE (FILEF) FASREC
000132 000 READ (ALTFL,2,END=3199) ALTTAB
000133 000 C END-IF
000134 000 GOTO 3990
000135 000 3199 ALTUIC = -1
000136 000 GOTO 3990
000137 000 C
000138 000 C *** COMPARE ABOVE-THE-LINE (UIC) WITH FAS (UIC) ***
000139 000 C
000140 000 3200 IF (ALTUIC *NE* FASUIC) GOTO 3400
000141 000 C THEN
000142 000 C CHECK FOR ATL-OVERAGE
000143 000 IF (ATLFLG *NE* '1') GOTO 3300
000144 000 C THEN
000145 000 C TREAT THIS RECORD AS AN OVERAGE
000146 000 TATLOV = TATLOV + 1
000147 000 AUTH13 = ZERO
000148 000 AUTH47 = ZERO
000149 000 AUTH8 = ZERO
000150 000 AUTH9A = NINES
000151 000 AUTHBD = NINES
000152 000 AUTHE = ZERO
000153 000 IF (MOD(COUNT,50) *EQ* 0) CALL PRTFAS (COUNT,2,FASTAB)
000154 000 GOTO 3350
000155 000 C ELSE
000156 000 C PROCESS ONE EACH ATL WITH CURRENT FAS REC
000157 000 3300 TALT = TALT + 1
000158 000 ROBCO = ALTTAB(3)
000159 000 AUTH13 = BLANK
000160 000 AUTH47 = BLANK
000161 000 AUTH8 = ONE
000162 000 AUTH9A = ALTTAB(5)
000163 000 AUTHBD = ALTTAB(6)
000164 000 LRE = ONE
000165 000 IF (FLGALT *NE* YES) GOTO 3325
000166 000 AUTH8 = FASTAB(45)
000167 000 AUTH9A = BLANK
000168 000 LRE = BLANK
000169 000 3325 CONTINUE
```


UNCLASSIFIED

```
000170 000      ENCODE (2,102,AUTHE) LRE, ATLTAB(4)
000171 000      IF (MOD(COUNT,50) .EQ. 0) CALL PRTFAS (COUNT,3,FASTAB)
000172 000      END-IF
000173 000      NEXT = FAS
000174 000      ENCODE (312,1,FASREC) FASTAB
000175 000      WRITE (FILEF) FASREC
000176 000      READ (ATLFIL,8,END=3399) ATLTAB
000177 000      END-IF
000178 000      GOTO 3980
000179 000      3399 ATLUIC = -1
000180 000      GOTO 3980
000181 000      C
000182 000      C *** COMPARE LOCKOUT (UIC) WITH FAS (UIC) ***
000183 000      C
000184 000      3400 IF (LOUIC .NE. FASUIC) GOTO 3500
000185 000      C      THEN
000186 000      C      CHECK LOCKOUT TYPE
000187 000      IF (LOTYPE .NE. COMPO) GOTO 3420
000188 000      C      THEN
000189 000      C      BY PASS THIS RECORD
000190 000      C      TBYPAS = TBYPAS + 1
000191 000      IF (MOD(COUNT,50) .EQ. 0) CALL PRTFAS (COUNT,4,FASTAB)
000192 000      C      GOTO 3440
000193 000      C      ELSE
000194 000      C      PROCESS ONE EACH LOCKOUT REC WITH CURRENT FAS REC
000195 000      C      3420 TLKOUT = TLKOUT + 1
000196 000      C      ROBCO = LOTAB(12)
000197 000      C      AUTH8 = X
000198 000      C      AUTH13 = BLANK
000199 000      C      AUTH47 = BLANK
000200 000      C      AUTH9A = BLANK
000201 000      C      AUTHBD = BLANK
000202 000      C      AUTHE = BLANK
000203 000      C      IF (MOD(COUNT,50) .EQ. 0) CALL PRTFAS (COUNT,5,FASTAB)
000204 000      C      ENCODE (312,1,FASREC) FASTAB
000205 000      C      WRITE (FILEF) FASREC
000206 000      C      END-IF
000207 000      C      3440 CONTINUE
000208 000      C
000209 000      END-IF
000210 000      NEXT = FAS
000211 000      READ (LOFILE,2,END=3499) LOTAB
000212 000      GOTO 3970
000213 000      3499 LOUIC = -1
000214 000      GOTO 3970
000215 000      C
000216 000      C *** COMPARE HAND-PLAYED (UIC) WITH FAS (UIC) ***
000217 000      C      3500 IF (HPUIC .NE. FASUIC) GOTO 3600
000218 000      C      THEN
000219 000      C      PROCESS ONE EACH HAND-PLAYED UNIT WITH CURRENT FAS REC
000220 000      C      THP = THP + 1
000221 000      C      ROBCO = HPTAB(13)
000222 000      C      AUTH8 = ONE
000223 000      C      AUTHBD = HPTAB(16)
000224 000      C      DECODE (3,4,HPTAB(6)) ROU
000225 000      C      4 FORMAT (13)
000226 000      C      5 FORMAT (J2)
```

UNCLASSIFIED

```
000227 000 CALL GETTP (MOD,TIMEPD)
000228 000 ENCODE (2,5,AUTH9A) TIMEPD
000229 000 ENCODE (2,102,AUTH8) BLANK, ONE
000230 000 AUTH13 = BLANK
000231 000 AUTH47 = BLANK
000232 000 IF (MOD(COUNT,50) *EQ. 0) CALL PRTFAS (COUNT,6,FASTAB)
000233 000 ENCODE (3,12,1,FASREC) FASTAB
000234 000 WRITE (FILEF) FASREC
000235 000 NEXT = FAS
000236 000 READ (HPFILE,8,END=3599) HPTAB
000237 000 C
000238 000 END-IF
000239 000 GOTO 3960
000240 000 3599 HPUC = -1
000241 000 GOTO 3960
000242 000 C
000243 000 C *** CHECK FOR NOTIONAL FROM MRFILE ***
000244 000 3600 IF (MUC *EQ. FASUC) GOTO 3650
000245 000 C THEN
000246 000 C PROCESS ONE EACH NOTIONAL UNIT
000247 000 IF (MRFLAG *NE. '0') GOTO 3995
000248 000 TSHORT = TSHORT + 1
000249 000 NEXT = NOT
000250 000 MRAB(29) = BLANK
000251 000 IF (MOD(COUNT,50) *EQ. 0) CALL PRTFAS (COUNT,7,NOTTAB)
000252 000 I = 1
000253 000 C
000254 000 C DOWNHILE (I *LE. 44)
000255 000 C
000256 000 3620 IF (I *GT. 44) GOTO 3640
000257 000 K = INDEX(I)
000258 000 NOTTAB(K) = MRAB(I)
000259 000 I = I + 1
000260 000 GOTO 3620
000261 000 C
000262 000 C END-DO
000263 000 3640 ENCODE (2,102,NOTTAB(14)) NOTTAB(14), ONE
000264 000 NOTTAB(82) = BLANK
000265 000 ENCODE (3,12,1,NOTREC) NOTTAB
000266 000 WRITE (FILEF) NOTREC
000267 000 GOTO 3700
000268 000 C
000269 000 C ELSE
000270 000 C THIS UNIT IS EITHER A MATCH-REQ UNIT OR AN OVERAGE
000271 000 3650 IF (MRFLAG *NE. '1') GOTO 3660
000272 000 C THEN
000273 000 C PROCESS ONE EACH MATCH-REQ WITH CURRENT FAS-REC
000274 000 TMATCH = TMATCH + 1
000275 000 ROBCO = MRAB(22)
000276 000 AUTH13 = MRAB(7)
000277 000 AUTH47 = MRAB(8)
000278 000 AUTH8 = MRAB(9)
000279 000 AUTH9A = MRAB(10)
000280 000 AUTHBD = MRAB(11)
000281 000 ENCODE (2,102,AUTH8) MRAB(12), ONE
000282 000 IF (MOD(COUNT,50) *EQ. 0) CALL PRTFAS (COUNT,8,FASTAB)
000283 000 C
000284 000 C
000285 000 C
000286 000 C
000287 000 C
000288 000 C
000289 000 C
000290 000 C
000291 000 C
000292 000 C
000293 000 C
000294 000 C
000295 000 C
000296 000 C
000297 000 C
000298 000 C
000299 000 C
000300 000 C
000301 000 C
000302 000 C
000303 000 C
000304 000 C
000305 000 C
000306 000 C
000307 000 C
000308 000 C
000309 000 C
000310 000 C
000311 000 C
000312 000 C
000313 000 C
000314 000 C
000315 000 C
000316 000 C
000317 000 C
000318 000 C
000319 000 C
000320 000 C
000321 000 C
000322 000 C
000323 000 C
000324 000 C
000325 000 C
000326 000 C
000327 000 C
000328 000 C
000329 000 C
000330 000 C
000331 000 C
000332 000 C
000333 000 C
000334 000 C
000335 000 C
000336 000 C
000337 000 C
000338 000 C
000339 000 C
000340 000 C
000341 000 C
000342 000 C
000343 000 C
000344 000 C
000345 000 C
000346 000 C
000347 000 C
000348 000 C
000349 000 C
000350 000 C
000351 000 C
000352 000 C
000353 000 C
000354 000 C
000355 000 C
000356 000 C
000357 000 C
000358 000 C
000359 000 C
000360 000 C
000361 000 C
000362 000 C
000363 000 C
000364 000 C
000365 000 C
000366 000 C
000367 000 C
000368 000 C
000369 000 C
000370 000 C
000371 000 C
000372 000 C
000373 000 C
000374 000 C
000375 000 C
000376 000 C
000377 000 C
000378 000 C
000379 000 C
000380 000 C
000381 000 C
000382 000 C
000383 000 C
000384 000 C
000385 000 C
000386 000 C
000387 000 C
000388 000 C
000389 000 C
000390 000 C
000391 000 C
000392 000 C
000393 000 C
000394 000 C
000395 000 C
000396 000 C
000397 000 C
000398 000 C
000399 000 C
000400 000 C
```

●●UNCLASSIFIED●●

```

000284 000 C
000285 000 3660
000286 000 ROBCO = TOVER + 1
000287 000 ROBCO = MRTAB(22)
000288 000 AUTH13 = ZERO
000289 000 AUTH47 = ZERO
000290 000 AUTH8 = ZERO
000291 000 AUTH9A = NINES
000292 000 AUTHBD = NINES
000293 000 AUTHC = ZERO
000294 000 IF (MOD(COUNT,50) .EQ. 0) CALL PRTFAS (COUNT,9,F,FASTAB)
000295 000 END-IF
000296 000 NEXT = FAS
000297 000 ENCODE (312,1,F,ASREC) FASTAB
000298 000 *WRITE (FILEF) FASREC
000299 000 C END-IF
000300 000 3700 CONTINUE
000301 000 CALL SRKET (MRREC,DUMMY,33799)
000302 000 DECODE (150,3,MRREC) MRTAB
000303 000 GOTO 3950
000304 000 3799 MUIC = -1
000305 000 C
000306 000 3950 CONTINUE
000307 000 C
000308 000 3960 CONTINUE
000309 000 C
000310 000 3970 CONTINUE
000311 000 C
000312 000 3980 CONTINUE
000313 000 C
000314 000 3990 CONTINUE
000315 000 GOTO 3000
000316 000 C END=00
000317 000 C
000318 000 3995 WRITE (6,99) LUIC, FASUIC
000319 000 99 FORMAT (' *** LUIC=',A6,' FASUIC = ',A6,' WAS NOT FOUND ***')
000320 000 CALL PRTFAS(-1,9,F,FASTAB)
000321 000 3999 CONTINUE
000322 000 TOTAL = TAL+TATLOV+TATL+TBYPAS+TLKOUT+THP+TSHORT+TMATCH+TOVER
000323 000 WRITE (6,101) TAL,TATLOV,TATL,TBYPAS,TLKOUT,THP,TSHORT,
000324 000 TMATCH,TOVER,TOTAL
000325 000 101 FORMAT(' TAL=',16,' TATLOV=',16,' TATL=',16,' TBYPAS=',16,
000326 000 ' TLKOUT=',16,' THP=',16,' TSHORT=',16,' TMATCH=',16,
000327 000 ' TOVER=',16,' TOTAL =',18,' *** LAYIN FINI ***')
000328 000 102 FORMAT (2A1)
000329 000 8 FORMAT (9X,A1,6X,A6,33X,A3,17X,A1,A2,A3,6X,A6,A3)
000330 000 ENDFILE 11
000331 000 ENDFILE 12
000332 000 ENDFILE 13
000333 000 ENDFILE 14
000334 000 ENDFILE 15
000335 000 ENDFILE 16
000336 000 ENDFILE 17
000337 000
000338 000
000339 000
000340 000

```

FMA PROGRAM LAYIN

••UNCLASSIFIED••

000341 000 STATUS = CSF (2, 'BRKPT 18.')

000342 000 STOP

000343 000 END

END ELT.

000344 FSLAY, PRTFAS

ELT007 RL1870 08/16-07:40:43-14.)

000001 001 C

000002 001 C

000003 001 C

000004 004 C

000005 001 C

000006 001 C

000007 004 C

000008 001 C

000009 001 C

000010 001 C

000011 001 C

000012 001 C

000013 001 C

000014 001 C

000015 001 C

000016 001 C

000017 001 C

000018 001 C

000019 003 C

000020 001 C

000021 002 C

000022 002 C

000023 001 C

000024 001 C

000025 001 C

000026 002 C

000027 001 C

000028 001 C

END ELT.

P R T F A S

SUBROUTINE PRTFAS (I,J,FASTAB)

IMPLICIT INTEGER (A-Z)

INTEGER FASTAB(99)

INTEGER REASON(9)

DATA REASON, 'ALT-TH', 'ATL-OV', 'ATL', 'LO-BYP',
'LCKOUT', 'HPUNIT', 'NOTION', 'MATCH', 'OVERAG' /

WRITE (18,1) REASON(J), (FASTAB(K), K=1,34)

WRITE (18,2) (FASTAB(K), K=35,70)

WRITE (18,3) (FASTAB(K), K=71,82)

WRITE (18,4) I, (FASTAB(K), K=83,99)

RETURN

1 FORMAT (1H, A6, 1X, 2A1, 1X, 2(A6, 1X), A1, 1X, A6, 1X, A4, 1X, A2, 1X,
• A3, 1X, A4, 1X, A1, 1X, A2, 1X, A3, 1X, 3(A2, 1X), A6, 1X, 2(A2, 1X), A5, 1X,
• A3, 1X, A2, 1X, A3, 1X, 2(A2, 1X), 5(A1, 1X), A3, 1X, A5, 1X, A3, 1X, A3)
2 FORMAT (1H, 3X, A2, 1X, A3, 1X, A2, 1X, A6, 1X, A3, 1X, A2, 1X, A1, 1X, A2, 1X,
• A1, 1X, A3, 1X, A1, 1X, A2, 1X, A3, 1X, 5(A1, 1X), A3, 1X, A1, 1X, A5, 1X,
• 2(A2, 1X), 4(A1, 1X), A5, 1X, A2, 1X, A6, 1X, A3, 1X, A5, 1X, A6, 1X, A5, 1X,
• A1, 1X, A5)
3 FORMAT (1H, 3X, 4(A1, 1X), A5, 1X, A4, 1X, A6, 1X, A3, 1X, 3A6, 1X, A3)
4 FORMAT (1H, 16, 2X, 15(A5, 1X), A4, 1X, A1)

000345 FMA POST-PROCESSOR PROGRAM BINCOPY

UNCLASSIFIED

L,0

SELT.L 24MONTAGNE.BINCOPY-SUM
ELT007 RL1B70 08/16-07:40:45-110,1
000001 006 IDENTIFICATION DIVISION.
000002 006 PROGRAM-ID. BINCOPY-A.
000003 006 AUTHOR. E MONTAGNE.
000004 006 DATE WRITTEN. 17 JULY 1975.
000005 006 REMARKS. THIS PROGRAM IS PART OF A SERIES OF NEW CAMP
000006 006 PROGRAMS. IT CALLS A FORTRAN SUBROUTINE TO READ AN
000007 006 UNFORMATTED BINARY FILE (BIFAS) INTO A COBOL FILE CALLED FASBINJ07.
000008 006 BIFAS IS CREATED BY KELLER TO CONTAIN RESULTS OF A MATCH OF
000009 006 CONWAY'S OUTPUT AGAINST THE FASTALS OUTPUT.
000010 006 FASBINJ07 IS THE FAS FILE (307 CHARACTERS PER RECORD, 24 RECORDS
000011 006 PER BLOCK) WHICH IS THEN SORTED AND OUTPUT TO TAPE (FAS-OUT FILE).
000012 006 SELECTED FIELDS ARE LISTED TO
000013 006 ASSURE CORRECTNESS PRIOR TO DELIVERY OF THE TAPE.
000014 006 REVISED 24 JULY TO REPLACE BLANKS WITH ZERO
000015 006 IN SELECTED FIELDS (AUTHR ESCON EQCUN).
000016 006 REVISED 15 AUG FOR TOTAL FORCE.
000017 006
000018 006 ENVIRONMENT DIVISION.
000019 006 CONFIGURATION SECTION.
000020 006 SOURCE-COMPUTER. UNIVAC-1106.
000021 006 OBJECT-COMPUTER. UNIVAC-1106.
000022 006 INPUT-OUTPUT SECTION.
000023 006 FILE-CONTROL.
000024 006 SELECT PRINT-FILE ASSIGN TO PRINTER.
000025 006 SELECT SORT-ING ASSIGN TO MASS-STORAGE XA.
000026 006 SELECT FAS-OUT ASSIGN TO UNISERVO TAPE-FILE.
000027 006 SELECT ALT-PRINT-FILE ASSIGN TO PRINTER 24FAS-PRINT.
000028 006
000029 006 DATA DIVISION.
000030 006 FILE SECTION.
000031 006 SD SORT-ING.
000032 006 U1 SORT-REC.
000033 006 02 FILLER PICTURE XX.
000034 006 02 COMPO PICTURE 9.
000035 006 02 UICCC PICTURE X(6).
000036 006 02 EDATE PICTURE 9(6).
000037 006 02 FILLER PICTURE X(13).
000038 006 02 AUTHR PICTURE X(15).
000039 006 02 FILLER PICTURE X(31).
000040 006 02 EQCON PICTURE X.
000041 006 02 ESCON PICTURE X.
000042 006 02 FILLER PICTURE X(231)
000043 006 FD ALT-PRINT-FILE LABEL RECORD OMITTED.
000044 006 U1 ALT-REC PIC X(132).
000045 006 FD FAS-OUT BLOCK 24 RECORDS LABEL RECORD OMITTED RECORDING
000046 006 MODE IS 1.
000047 006 U1 FAS-REC.
000048 006 02 FILLER PICTURE X.
000049 006 02 FICOD PICTURE X.
000050 006 02 COMPO PICTURE X.
000051 006 02 UICCC PICTURE X(6).
000052 006 02 EDATE PICTURE 9(6).
000053 006 02 FILLER PICTURE X(13).
000054 006 02 AUTHR.
000055 006 U3 UIN PICTURE XXX.
000056 006 U3 FLAGS PICTURE XXXX.

000056	006	03	MATCH-CODE	PIC X.
000057	006	03	TP	PIC XX.
000058	006	03	RDD	PIC XXX.
000059	006	03	LRE	PIC X.
000060	006	03	TH-CODE	PICTURE X.
000061	006	02	BRNCH	PIC XX.
000062	006	02	CARS	PIC XX.
000063	006	02	FILLER	PIC X(23).
000064	006	02	OSCMF	PIC XX.
000065	006	02	FILLER	PIC XX.
000066	006	02	EGCON	PIC X.
000067	006	02	ESCON	PIC X.
000068	006	02	FILLER	PIC XXX.
000069	006	02	FPLAN	PIC XXX.
000070	006	02	JCSTY	PIC X(5).
000071	006	02	LOCCO	PIC X(3).
000072	006	02	FILLER	PIC X(39).
000073	006	02	ROBCO	PIC XXX.
000074	006	02	FILLER	PIC X.
000075	006	02	SRCIO	PIC X(13).
000076	006	02	FILLER	PIC X(7).
000077	006	02	STNNM	PIC X(9).
000078	006	02	FILLER	PIC X(17).
000079	006	02	TPSNA	PIC X(5).
000080	006	02	FILLER	PIC X(9).
000081	006	02	UNHBR	PIC X(4).
000082	006	02	PECOD	PIC X(6).
000083	006	02	ULCCC	PIC X(3).
000084	006	02	UNTDS-18	PIC X(18).
000085	006	02	FILLER	PIC X(3).
000086	006	02	AUOFF	PIC X(5).
000087	006	02	AUWOF	PIC X(5).
000088	006	02	AUENL	PIC X(5).
000089	006	02	AUAGR	PIC X(5).
000090	006	02	AUCIV	PIC X(5).
000091	006	02	FILLER	PIC X(15).
000092	006	02	OPAGR	PIC X(5).
000093	006	02	FILLER	PIC X(5).
000094	006	02	STOFF	PIC X(5).
000095	006	02	STWOF	PIC X(5).
000096	006	02	STENL	PIC X(5).
000097	006	02	STAGR	PIC 9(5).
000098	006	02	STCIV	PIC X(5).
000099	006	02	FILLER	PIC X(5).
000100	006	FD	PRINT-FILE	LABEL RECORD OMITTED.
000101	006	01	LINE-REC-3	PIC X(132).
000102	006	WORKING-STORAGE	SECTION.	
000103	006	77 K	PIC 99	USAGE IS COMPUTATIONAL.
000104	006	77 J	PIC 99	USAGE IS COMPUTATIONAL.
000105	006	77 I	PIC 99	USAGE IS COMPUTATIONAL.
000106	006	77	TIME-PERIOD	PIC 99.
000107	006	77	COMPO-CODE	PIC X.
000108	006	77	COMPO-CODE-1	PIC 9.
000109	006	01	STAGR-TABLE.	
000110	006	02	STAGR-TP	OCCURS 10 TIMES.
000111	006	03	STAGR-SUM	OCCURS 9 TIMES PIC 999999.
000112	006	01	SUM-TABLE.	

UNCLASSIFIED

```
000113 006 02 CUM-STAGR-TP OCCURS 10 TIMES.
000114 006 03 CUM-STAGR-SUM OCCURS 9 TIMES PIC 9(7).
000115 006 77 COUNT-REC PICTURE 9(5) USAGE IS COMPUTATIONAL VALUE ZERO.
000116 006 77 LINE-NR PIC 99 USAGE IS COMPUTATIONAL.
000117 006 77 IEND-OF-FILE PICTURE 9 VALUE 2.
000118 006 77 FASBIN-REC-312 PICTURE X(312).
000119 006 01 LINE-REC VALUE SPACES.
000120 006 02 SEQNR PIC 9(5).
000121 006 02 FILLER PIC X.
000122 006 02 COMPO PIC X.
000123 006 02 FILLER PIC X(5).
000124 006 02 UICCC PIC X(6).
000125 006 02 FILLER PIC X.
000126 006 02 EDATC PIC 9(6).
000127 006 02 FILLER PIC X.
000128 006 02 SRCIO PIC X(13).
000129 006 02 FILLER PIC X.
000130 006 02 TPSNA PIC X(5).
000131 006 02 FILLER PIC X.
000132 006 02 FPLAN PIC X(3).
000133 006 02 FILLER PIC X(5).
000134 006 02 CARSS PIC XX.
000135 006 02 FILLER PIC X.
000136 006 02 UNMBR PIC X(4).
000137 006 02 FILLER PIC XX.
000138 006 02 BRNCH PIC X(2).
000139 006 02 FILLER PIC X.
000140 006 02 ULCCC PIC X(3).
000141 006 02 FILLER PIC X.
000142 006 02 UNTDS-18 PIC X(18).
000143 006 02 FILLER PIC X.
000144 006 02 STNMH PIC X(9).
000145 006 02 FILLER PIC X.
000146 006 02 LOCCO PIC X(3).
000147 006 02 FILLER PIC XXX.
000148 006 02 AUTHR.
000149 006 03 VIN PIC XXX.
000150 006 03 FILLER PIC X.
000151 006 03 FLAGS PIC XXXX.
000152 006 03 FILLER PIC X.
000153 006 03 MATCH-CODE PIC X.
000154 006 03 FILLER PIC X.
000155 006 03 TP PIC XX.
000156 006 03 FILLER PIC X.
000157 006 03 RDD PIC XXX.
000158 006 03 FILLER PIC X.
000159 006 03 LRE PIC X.
000160 006 03 FILLER PIC X.
000161 006 03 TH-CODE PIC X.
000162 006 02 FILLER PIC X.
000163 006 02 FILLER PIC XX.
000164 006 02 FILLER PIC XXX.
000165 006 01 LINE-REC-2 VALUE SPACES.
000166 006 02 FILLER PIC X(11).
000167 006 02 USCMP PIC XX.
000168 006 02 FILLER PIC X(4).
000169 006 02 JCSTY PIC X(5).
```

UNCLASSIFIED

•UNCLASSIFIED•

FMA POST-PROCESSOR PROGRAM BINCOPY

```

000170 006 02 FILLER PIC X.
000171 006 02 ROBCO PIC XXX.
000172 006 02 FILLER PIC XXX.
000173 006 02 PECOD PIC X(6).
000174 006 02 FILLER PIC X.
000175 006 02 STAGR PIC X(5).
000176 006 02 FILLER PIC X(91).
000177 006 01 FASBIN-REC-307.
000178 006 02 FILLER PICTURE X.
000179 006 02 FICOD PICTURE X.
000180 006 02 COMPO PICTURE 9.
000181 006 02 UICCC PICTURE X(6).
000182 006 02 EDATE PICTURE 9(6).
000183 006 02 FILLER PICTURE X(13).
000184 006 02 AUTHR PIC X(15).
000185 006 02 FILLER PICTURE X(31).
000186 006 02 EQCON PICTURE X.
000187 006 02 ESCON PICTURE X.
000188 006 02 FILLER PICTURE X(151).
000189 006 02 STRENGTHS PIC X(75).
000190 006 02 FILLER PIC X(5).
000191 006 01 HEADER-1.
000192 006 02 FILLER PIC X(89) VALUE 'SEQNR COMPO UICCC EDATE SRCTO -
000193 006 - - - -TPSNA FPLAN CA UNMBR BR ULC UNITS - - - -'.
000194 006 02 FILLER PIC X(35) VALUE 'STNNM LOCCO AUTHR - - - -'.
000195 006 - - - -'.
000196 006 02 FILLER PIC X(8) VALUE SPACES.
000197 006 01 HEADER-2.
000198 006 02 FILLER PIC X(44) VALUE ' DSC JCSTY ROBCO
000199 006 - 'PECOD STAGR'.
000200 006 02 FILLER PIC X(88) VALUE SPACES.
000201 006 PROCEDURE DIVISION.
000202 006 OPEN-UP SECTION.
000203 007 OPEN OUTPUT FAS-OUT PRINT-FILE ALT-PRINT-FILE.
000204 006 FAS-SORT SECTION.
000205 006 SORT SORT-ING ON ASCENDING KEY COMPO UICCC EDATE
000206 006 INPUT PROCEDURE IS ENTER-FORTRAN
000207 006 OUTPUT PROCEDURE IS WRITE-TAPE.
000208 006 GO TO END-RUN.
000209 006 ENTER-FORTRAN SECTION.
000210 006 ENTER FORTRAN READ81 SUBROUTINE REFERENCING
000211 006 FASBIN-REC-312 IEND-OF-FILE.
000212 006 IF IEND-OF-FILE = 4 GO TO CLOSE-307.
000213 006 ADD 1 TO COUNT-REC.
000214 006 MOVE FASBIN-REC-312 TO FASBIN-REC-307.
000215 006 EXAMINE STRENGTHS REPLACING ALL SPACES BY ZERO.
000216 006 RELEASE SORT-REC FROM FASBIN-REC-307.
000217 006 GO TO ENTER-FORTRAN.
000218 006 CLOSE-307.
000219 006 MOVE SPACES TO LINE-REC MOVE COUNT-REC TO SEVNR.
000220 006 MOVE 'RECORDS ' TO SRCTO IN LINE-REC.
000221 006 WRITE LINE-REC-3 FROM LINE-REC.
000222 006 MOVE ZERO TO SEVNR.
000223 006 WRITE-TAPE SECTION.
000224 008 PERFORM SET-ZERO VARYING J FROM 1 BY 1 UNTIL J = 11.
000225 006 SET-HEADER.
000226 006 MOVE ZERO TO LINE-NR.

```

•UNCLASSIFIED•

FMA POST-PROCESSOR PROGRAM BINCOPY


```
000227 007 WRITE ALT-REC FROM HEADER-1 AFTER ADVANCING TOP-OF-PAGE.
000228 007 WRITE ALT-REC FROM HEADER-2 MOVE SPACES TO LINE-REC-3.
000229 007 WRITE ALT-REC.
000230 006 WRITE-TAPE-LINES.
000231 006 RETURN SORT-ING RECORD INTO FAS-REC AT END GO TO PARA-1.
000232 009 IF TP IN FAS-REC NOT NUMERIC OR COMPO IN FAS-REC
000233 009 NOT NUMERIC GO TO SKIP-COMP.
000234 006 MOVE TP IN FAS-REC TO TIME-PERIOD.
000235 009 IF TIME-PERIOD < 1 OR > 8 GO TO SKIP-COMP.
000236 009 MOVE COMPO IN FAS-REC TO COMPO-CODE-1.
000237 006 ADD STAGR IN FAS-REC TO
000238 006 STAGR-SUM(TIME-PERIOD, COMPO-CODE-1).
000239 009 SKIP-COMP.
000240 006 ADD 1 TO SEQNR.
000241 006 MOVE CORRESPONDING FAS-REC TO LINE-REC.
000242 007 WRITE ALT-REC FROM LINE-REC.
000243 006 MOVE CORRESPONDING FAS-REC TO LINE-REC-2.
000244 007 WRITE ALT-REC FROM LINE-REC-2 WRITE FAS-REC.
000245 006 ADD 2 TO LINE-NR.
000246 006 IF LINE-NR = 54 GO TO SET-HEADER.
000247 006 GO TO WRITE-TAPE-LINES.
000248 006 PARA-1.
000249 006 EXIT.
000250 006 END-RUN SECTION.
000251 006 PERFORM COMPUTE-SUM VARYING I FROM 1 BY 1 UNTIL
000252 006 I = 11.
000253 006 PERFORM COMP-CUM-SUM VARYING J FROM 2 BY 1
000254 006 UNTIL J = 11.
000255 006 CLOSE PRINT-FILE.
000256 007 CLOSE FAS-OUT ALT-PRINT-FILE.
000257 006 STOP RUN.
000258 006 SET-ZERO.
000259 006 PERFORM SET-1 VARYING I FROM 1 BY 1 UNTIL I = 10.
000260 006 SET-1.
000261 006 COMPUTE STAGR-SUM(J, 1) = 0.
000262 006 COMPUTE CUM-STAGR-SUM(J, 1) = 0.
000263 006 COMPUTE-SUM.
000264 009 COMPUTE STAGR-SUM(I, 6) = STAGR-SUM(I, 2) +
000265 009 STAGR-SUM(I, 3).
000266 009 COMPUTE STAGR-SUM(I, 7) = STAGR-SUM(I, 6) +
000267 009 STAGR-SUM(I, 4).
000268 009 COMPUTE STAGR-SUM(I, 8) = STAGR-SUM(I, 1) +
000269 009 STAGR-SUM(I, 6).
000270 009 COMPUTE STAGR-SUM(I, 9) = STAGR-SUM(I, 7) +
000271 009 STAGR-SUM(I, 1).
000272 006 MOVE STAGR-TP(I) TO LINE-REC-3 WRITE LINE-REC-3.
000273 006 COMP-CUM-SUM.
000274 006 PERFORM COMP-CUM-SUM-1 VARYING I FROM 1 BY 1 UNTIL
000275 006 I = 10.
000276 006 MOVE CUM-STAGR-TP(J) TO LINE-REC-3 WRITE
000277 006 LINE-REC-3.
000278 006 COMP-CUM-SUM-1.
000279 006 COMPUTE K = J - 1 COMPUTE CUM-STAGR-SUM(J, 1) =
000280 010 CUM-STAGR-SUM(K, 1) + STAGR-SUM(K, 1).
```

END ELT.

```

000001 025 24MONTAGNE,READBIN
000002 025 ELT007 RL1870 08/16-07:40:48-(29.1)
000003 025 SUBROUTINE READBI (FASBIN,IEND)
000004 025 DATA IFILE/ICOUNT/0/KCOUNT/0/LCOUNT/0/
000005 025 DIMENSION FASBIN(52)
000006 029 C THIS SUBROUTINE READS A BINARY UNFORMATTED FORTRAN FILE.
000007 025 C THE FILE CONTAINS FAS RECORDS WITH MATCH RESULTS.
000008 025 C RECORDS ARE 52 WORDS (312 CHARACTERS). READBIN IS MEANT TO
000009 025 C BE CALLED BY A COBOL PROGRAM.
000010 025 C DO STATEMENT FOR DEBUG ONLY.
000011 025 C DO 200 I = 1, 50
000012 029 200 READ (IFILE,IEND=30) FASBIN
000013 025 KCOUNT = KCOUNT + 1
000014 025 RETURN
000015 025 120 FORMAT (10X,15,5X,15)
000016 025 199 IEND = 1H4
000017 025 30 PRINT 199, KCOUNT
000018 025 RETURN
000019 025 END

```

END ELT.

BHGG MRG PRE-PROCESSOR PROGRAM SORTARLOC ••UNCLASSIFIED•• L.D

```
WELT.L 24MONTAGNE.ARLCORT
ELT007 RL1870 08/16-07:40:49-11.)
000001 000 IDENTIFICATION DIVISION.
000002 000 PROGRAM-ID. ARLCORT.
000003 000 AUTHOR. E MONTAGNE.
000004 000 DATE-WRITTEN. 9 JULY 1975.
000005 000 REMARKS. INPUT IS ARLC TAPE FROM AOC. TAPE CONTENTS ARE
000006 000 PRINTED, THEN SORTED BY LOCNA, AREA, AND TLAC. THE SORTED
000007 000 FILE IS PRINTED AND OUTPUT TO TAPE.
000008 000 ENVIRONMENT DIVISION.
000009 000 CONFIGURATION SECTION.
000010 000 SOURCE-COMPUTER. UNIVAC-1108.
000011 000 OBJECT-COMPUTER. UNIVAC-1108.
000012 000 INPUT-OUTPUT SECTION.
000013 000 FILE-CONTROL.
000014 000 SELECT AFTER-SORT ASSIGN TO UNISERVO TAPE-SORT.
000015 000 SELECT SORT-ING ASSIGN TO MASS-STORAGE XA.
000016 000 SELECT PRINT-FILE ASSIGN TO PRINTER.
000017 000 SELECT TAPE-FILE ASSIGN TO UNISERVO TAPE-FILE.
000018 000 DATA DIVISION.
000019 000 FILE SECTION.
000020 000 SD SORT-ING.
000021 000 01 SORTGOREC.
000022 000 02 AREA PICTURE X(13).
000023 000 02 FILLER PICTURE X(15).
000024 000 02 ARLC PICTURE X(5).
000025 000 02 FILLER PICTURE X(4).
000026 000 02 GELOC PICTURE X(4).
000027 000 02 FILLER PICTURE X(15).
000028 000 02 LOCNA PICTURE X(9).
000029 000 02 FILLER PICTURE X(4).
000030 000 02 LOCNM PICTURE X(17).
000031 000 02 FILLER PICTURE X(8).
000032 000 02 TLAC PICTURE X(3).
000033 000 02 FILLER PICTURE X(10).
000034 000 02 FCTCD PICTURE X(18).
000035 000 02 FILLER PICTURE X(2).
000036 000 02 UGRID PICTURE X(11).
000037 000 02 FILLER PICTURE X(3).
000038 000 02 LFLC PICTURE X(15).
000039 000 02 FILLER PICTURE X(2).
000040 000 02 STAT PICTURE X(1).
000041 000 02 FILLER PICTURE X(3).
000042 000 FD AFTER-SORT LABEL RECORD OMITTED.
000043 000 01 SORTED-REC PICTURE X(132).
000044 000 FD PRINT-FILE LABEL RECORD OMITTED.
000045 000 01 PRINT-LINE PICTURE X(132).
000046 000 FD TAPE-FILE LABEL RECORD OMITTED RECORDING MODE IS 1.
000047 000 01 TAPE-REC PICTURE X(132).
000048 000 WORKING-STORAGE SECTION.
000049 000 01 HEADER-LINE.
000050 000 02 FILLER PICTURE X(26) VALUE 'AREA' ARLC GELOC ' '
000051 000 02 FILLER PICTURE X(26) VALUE 'LOCNA' LOCNM ' '
000052 000 02 FILLER PICTURE X(25) VALUE ' ' TLAC ' '
000053 000 02 FILLER PICTURE X(25) VALUE 'FCTCD' UGRID ' '
000054 000 02 FILLER PICTURE X(26) VALUE ' ' LFLC ' '
000055 000 02 FILLER PICTURE X(4) VALUE 'STAT' ' '

```


UNCLASSIFIED

MRG PRE-PROCESSOR PROGRAM SORTARLOC

```

000056 000 01 TOP-HEADER.
000057 000 02 FILLER PICTURE X155) VALUE SPACES.
000058 000 02 FILLER PICTURE X123) VALUE 'DA GEOGRAPHIC LOCATIONS'.
000059 000 02 FILLER PICTURE X141) VALUE SPACES.
000060 000 02 FILLER PICTURE X151) VALUE 'PAGE'.
000061 000 02 PAGENR PICTURE 999 VALUE 0.
000062 000 02 FILLER PICTURE X151) VALUE SPACES.
000063 000 01 LAST-LINE.
000064 000 02 FILLER PICTURE X124) VALUE ' NUMBER OF RECORDS IS '.
000065 000 02 0-REC-COUNT PICTURE ZZZZ.
000066 000 02 FILLER PICTURE X103) VALUE SPACES.
000067 000 77 LINE-COUNT PICTURE 99.
000068 000 77 REC-COUNT PICTURE 99999 VALUE ZERO.
000069 000 PROCEDURE DIVISION.
000070 000 OPEN-FILES.
000071 000 OPEN INPUT TAPE-FILE.
000072 000 OPEN OUTPUT PRINT-FILE.
000073 000 OPEN OUTPUT AFTER-SORT.
000074 000 A-SORT SECTION.
000075 000 SORT SORT-ING ON ASCENDING KEY LOCNA AREA TLAC
000076 000 INPUT PROCEDURE IS I-SORT
000077 000 OUTPUT PROCEDURE IS 0-SORT.
000078 000 GO TO PRINT-SORTED.
000079 000 I-SORT SECTION.
000080 000 SET-HEADER.
000081 000 ADD 1 TO PAGENR MOVE 1 TO LINE-COUNT.
000082 000 WRITE PRINT-LINE FROM TOP-HEADER AFTER ADVANCING
000083 000 TOP-OF-PAGE.
000084 000 MOVE SPACES TO PRINT-LINE WRITE PRINT-LINE.
000085 000 WRITE PRINT-LINE FROM HEADER-LINE.
000086 000 MOVE SPACES TO PRINT-LINE WRITE PRINT-LINE.
000087 000 READ-FILE.
000088 000 READ TAPE-FILE AT END GO TO END-I-SORT.
000089 000 ADD 1 TO REC-COUNT WRITE PRINT-LINE FROM TAPE-REC.
000090 000 RELEASE SORTGOREC FROM TAPE-REC.
000091 000 IF LINE-COUNT EQUAL TO 53 GO TO SET-HEADER.
000092 000 ADD 1 TO LINE-COUNT GO TO READ-FILE.
000093 000 END-I-SORT.
000094 000 CLOSE TAPE-FILE MOVE REC-COUNT TO 0-REC-COUNT.
000095 000 WRITE PRINT-LINE FROM LAST-LINE.
000096 000 0-SORT SECTION.
000097 000 RETURN SORT-ING RECORD INTO SORTED-REC AT END GO TO
000098 000 END-0-SORT.
000099 000 WRITE SORTED-REC GO TO 0-SORT.
000100 000 END-0-SORT.
000101 000 CLOSE AFTER-SORT.
000102 000 PRINT-SORTED SECTION.
000103 000 OPEN INPUT AFTER-SORT.
000104 000 MOVE 0 TO PAGENR.
000105 000 MOVE 0 TO REC-COUNT.
000106 000 RESET-HEADER.
000107 000 ADD 1 TO PAGENR MOVE 1 TO LINE-COUNT.
000108 000 WRITE PRINT-LINE FROM TOP-HEADER AFTER ADVANCING
000109 000 TOP-OF-PAGE.
000110 000 WRITE PRINT-LINE FROM HEADER-LINE AFTER 2.
000111 000 MOVE SPACES TO PRINT-LINE WRITE PRINT-LINE.
000112 000 READ-SORTED-FILE.

```

UNCLASSIFIED

MRG PRE-PROCESSOR PROGRAM SORTARLOC


```
000113 000 READ AFTER-SORT AT END GO TO END-RUN*
000114 000 ADD 1 TO REC-COUNT WRITE PRINT-LINE FROM SORTED-REC*
000115 000 IF LINE-COUNT EQUAL TO 53 GO TO RESET-HEADER*
000116 000 ADD 1 TO LINE-COUNT GO TO READ-SORTED-FILE*
000117 000 END-RUN*
000118 000 MOVE REC-COUNT TO 0-REC-COUNT*
000119 000 WRITE PRINT-LINE FROM LAST-LINE*
000120 000 CLOSE AFTER-SORT PRINT-FILE*
000121 000 STOP RUN*
```

END ELT.

```
SELTL 24MONTAGNE*ARLOC=23
ELT007 RL1B70 06/16-07:40:51-(14.)
000001 010 IDENTIFICATION DIVISION.
000002 010 PROGRAM-ID. ARLOC-23.
000003 010 AUTHOR. E MONTAGNE.
000004 010 DATE-WRITTEN. 12 SEPT 75.
000005 010 REMARKS. PROGRAM SORTS SELECTED FIELDS OF ARLOC FILE AND
000006 010 PUTS RESULTS IN FILE 24SORTARLOC.
000007 010 ENVIRONMENT DIVISION.
000008 010 CONFIGURATION SECTION.
000009 010 SOURCE-COMPUTER. UNIVAC-1108.
000010 010 OBJECT-COMPUTER. UNIVAC-1108.
000011 010 INPUT-OUTPUT SECTION.
000012 010 FILE-CONTROL.
000013 010 SELECT OUT-FILE ASSIGN TO PRINTER 24SORTARLOC.
000014 010 SELECT SORT-ING ASSIGN TO MASS-STORAGE XA.
000015 010 SELECT PRINT-FILE ASSIGN TO PRINTER.
000016 010 SELECT ARLOC-FILE ASSIGN TO UNISERVU ARLOC-FILE.
000017 010 DATA DIVISION.
000018 010 FILE SECTION.
000019 010 SD SORT-ING.
000020 010 SI SORTGEOREC.
000021 011 02 AREA-AREA PIC X(13).
000022 010 02 LOCNA PIC X(9).
000023 010 02 GELOC PIC X(4).
000024 010 02 ARLOC PIC X(5).
000025 014 02 SAVE-AREA-1 PIC X.
000026 010 02 TLAC PIC X(3).
000027 010 FD ARLOC-FILE LABEL RECORD OMITTED RECORDING MODE IS 1.
000028 010 01 ARLOC-REC.
000029 011 02 AREA-AREA PIC X(13).
000030 010 02 FILLER PIC X(5).
000031 010 02 ARLOC PIC X(5).
000032 010 02 FILLER PIC X(4).
000033 010 02 GELOC PIC X(4).
000034 010 02 FILLER PIC X(5).
000035 010 02 LOCNA PIC X(9).
000036 010 02 FILLER PIC X(4).
000037 010 02 LOCNM PIC X(17).
000038 010 02 FILLER PIC X(18).
000039 010 02 TLAC PIC X(3).
000040 010 02 FILLER PIC X(10).
000041 010 02 FCTCD PIC X(18).
```

```

000042 010 02 FILLER PIC X(12).
000043 010 02 UGRID PIC X(11).
000044 010 02 FILLER PIC X(13).
000045 010 02 LFLC PIC X(15).
000046 010 02 FILLER PIC X(12).
000047 010 02 STAT PIC X(11).
000048 010 02 FILLER PIC X(13).
000049 010 FD OUT-FILE LABEL RECORD OMITTED.
000050 010 01 OUT-REC.
000051 011 02 AREA-AREA PIC X(3).
000052 010 02 LOCNA PIC X(9).
000053 010 02 GELOC PIC X(4).
000054 010 02 ARLOC PIC X(5).
000055 011 02 SAVE-AREA-1 PIC X.
000056 010 FD PRINT-FILE LABEL RECORD OMITTED.
000057 010 01 PRINT-LINE PIC X(132).
000058 010 WORKING-STORAGE SECTION.
000059 010 01 SORT-REC.
000060 011 02 AREA-AREA.
000061 011 03 AREA-1 PIC X.
000062 011 03 AREA2-3 PIC XX.
000063 010 02 LOCNA PIC X(9).
000064 010 02 GELOC PIC X(4).
000065 010 02 ARLOC PIC X(5).
000066 014 02 SAVE-AREA-1 PIC X.
000067 014 02 TLAC PIC X(3).
000068 010 01 LAST-LINE.
000069 010 02 FILLER PIC XX VALUE SPACES.
000070 010 02 0-REC-COUNT PIC ZZZZ.
000071 010 02 FILLER PIC X(9) VALUE IS ' RECORDS'.
000072 010 02 FILLER PIC X(116) VALUE SPACES.
000073 010 01 TEMP-TABLE.
000074 010 02 TEMP OCCURS 6 TIMES.
000075 011 03 AREA-AREA PIC X(3).
000076 010 03 LOCNAME PIC X(9).
000077 010 03 GELOC PIC X(4).
000078 010 03 ARLOC PIC X(5).
000079 012 03 SAVE-AREA-1 PIC X.
000080 014 03 TLAC PIC X(3).
000081 010 77 KAIN PIC 9 USAGE IS COMPUTATIONAL VALUE ZERO.
000082 010 77 KCTY PIC 9 USAGE IS COMPUTATIONAL VALUE ZERO.
000083 010 77 I PIC 99 USAGE IS COMPUTATIONAL VALUE ZERO.
000084 010 77 J PIC 99 USAGE IS COMPUTATIONAL.
000085 010 77 REC-COUNT PIC 99999 USAGE IS COMPUTATIONAL VALUE ZERO.
000086 010 PROCEDURE DIVISION.
000087 010 OPEN INPUT ARLOC-FILE OPEN OUTPUT OUT-FILE PRINT-FILE.
000088 010 A-SORT SECTION.
000089 011 SORT SORT-ING ON ASCENDING KEY AREA-AREA LOCNA
000090 010 INPUT PROCEDURE IS I-SORT
000091 010 OUTPUT PROCEDURE IS O-SORT.
000092 010 GO TO END-RUN.
000093 010 I-SORT SECTION.
000094 010 READ-FILE.
000095 010 READ ARLOC-FILE AT END GO TO END-I-SORT.
000096 010 ADD 1 TO REC-COUNT.
000097 010 MOVE CORRESPONDING ARLOC-REC TO SORT-REC.
000098 012 IF AREA-1 IN SORT-REC IS NUMERIC PERFORM

```

```

000099 012 SCRATCH=NUMERIC.
000100 010 RELEASE SORTGEOREC FROM SORT-REC.
000101 010 GO TO READ-FILE.
000102 012 SCRATCH=NUMERIC.
000103 013 MOVE AREA=1 IN SORT-REC TO SAVE-AREA=1 IN SORT-REC.
000104 012 MOVE SPACE TO AREA=1 IN SORT-REC.
000105 010 END=1-SORT.
000106 010 MOVE REC-COUNT TO O-REC-COUNT.
000107 010 WRITE PRINT-LINE FROM LAST-LINE.
000108 010 MOVE ZERO TO REC-COUNT MOVE SPACES TO PRINT-LINE.
000109 010 O-SORT SECTION.
000110 010
000111 010 • ELIMINATE DUPLICATE LOCNA RECORDS. •
000112 010
000113 010 CHECK=DUP.
000114 010 ADD 1 TO 1.
000115 010 RETURN SORT-ING RECORD INTO TEMP(1) AT END GO TO END-O-SORT.
000116 010 IF LOCNAME(1) = LOCNAME(1) GO TO CHECK-DUP.
000117 010 PERFORM SELECT-LOCNA THRU LAST-SEL.
000118 010 MOVE TEMP(1) TO TEMP(1) MOVE 1 TO 1 GO TO CHECK-DUP.
000119 010
000120 010 • SELECT LOCNA = AIN OR CTY IF POSSIBLE,
000121 010 • ELSE ARBITRARILY SELECT TEMP(1).
000122 010
000123 010 SELECT-LOCNA.
000124 010 PERFORM SELECT-AIN-CTY VARYING J FROM 1 BY 1 UNTIL J = 1.
000125 010 SELECT-AIN.
000126 010 IF KAIN EQUAL TO ZERO GO TO SELECT-CTY.
000127 010 MOVE TEMP(KAIN) TO OUT-REC GO TO LAST-SEL.
000128 010 SELECT-CTY.
000129 010 IF KCTY = ZERO GO TO NEXT-TO-LAST.
000130 010 MOVE TEMP(KCTY) TO OUT-REC GO TO LAST-SEL.
000131 010 SELECT-AIN-CTY.
000132 010 IF TRACE(J) = 'AIN' COMPUTE KAIN = J.
000133 010 IF TRACE(J) = 'CTY' COMPUTE KCTY = J.
000134 010 NEXT-TO-LAST.
000135 010 MOVE TEMP(1) TO OUT-REC.
000136 010 LAST-SEL.
000137 010 MOVE ZERO TO KCTY, KAIN.
000138 010 WRITE OUT-REC ADD 1 TO REC-COUNT WRITE PRINT-LINE
000139 010 FROM OUT-REC.
000140 010 END-O-SORT.
000141 010 PERFORM SELECT-LOCNA THRU LAST-SEL.
000142 010 MOVE REC-COUNT TO O-REC-COUNT.
000143 010 WRITE PRINT-LINE FROM LAST-LINE.
000144 010 END-RUN SECTION.
000145 010 CLOSE PRINT-FILE OUT-FILE ARLOC-FILE.
000146 010 STOP RUN.

```

END ELT.

MRG PRE-PROCESSOR PROGRAM MORSAROLL

ELT007 RL1870 08/16-07:40:54-117.)

```

000001 013 IDENTIFICATION DIVISION.
000002 013 PROGRAM-ID.
000003 013 AUTHOR.
000004 013 DATE-WRITTEN.
000005 013 REMARKS. PROGRAM HEADS MORSAR FILE. ELIMINATES ARMY RECORDS
000006 013 AND INTRA-THEATER RECORDS. SORTS ON RUD, DESTINATION,
000007 013 ORIGIN, TRAVEL MODE AND AVAIL DATE. THE REMAINING
000008 013 RECORDS ARE PACKAGED TO PRODUCE SMOBSMOD CARDS.
000009 013 ENVIRONMENT DIVISION.
000010 013 CONFIGURATION SECTION.
000011 013 SOURCE-COMPUTER. UNIVAC-1108.
000012 013 OBJECT-COMPUTER. UNIVAC-1108.
000013 013 INPUT-OUTPUT SECTION.
000014 013 FILE-CONTROL.
000015 013 SELECT PACKAGE-FILE ASSIGN TO PRINTER 24PACKAGE.
000016 013 SELECT MORSAR-FILE ASSIGN TO UNISERVU 24MORSAR-TP.
000017 013 SELECT PRINT-FILE ASSIGN TO PRINTER.
000018 013 SELECT 0-NODE-FILE ASSIGN TO CARD-HEADER.
000019 013 SELECT 0-NODE-FILE ASSIGN TO CARD-HEADER.
000020 013 SELECT SORT-ING ASSIGN TO MASS-STORAGE XA.
000021 013 SELECT CARD-FILE ASSIGN TO CARD-HEADER.
000022 013 DATA DIVISION.
000023 013 FILE SECTION.
000024 013 FD CARD-FILE LABEL RECORD OMITTED.
000025 013 01 CARD-IN PIC X(180).
000026 013 FD PACKAGE-FILE LABEL RECORD OMITTED.
000027 013 01 PACK-REC PIC X(132).
000028 013 FD MORSAR-FILE BLOCK 6 RECORDS RECORDING MODE 1
000029 013 LABEL RECORD OMITTED.
000030 013 01 MORSAR-REC.
000031 013 02 CNTL.
000032 013 03 PLAN-ID PIC X.
000033 013 03 SERV PIC X.
000034 013 03 SEQ PIC 9(4).
000035 013 03 TT PIC X.
000036 013 02 AGG-SEQ PIC X(4).
000037 013 02 RT PIC X.
000038 013 02 RD PIC X.
000039 013 02 REC-NAME PIC X(12).
000040 013 02 LEVEL PIC X(3).
000041 013 02 T-MODE.
000042 013 03 T-MODE1 PIC X.
000043 013 03 T-MODE2-3 PIC X.
000044 013 02 AVAIL PIC 999.
000045 013 02 RUD PIC 999.
000046 013 02 PORT-DATE PIC X(3).
000047 013 02 NODE-DATE PIC X(3).
000048 013 02 ORIGIN-SET.
000049 013 03 NAME=0 PIC X(14).
000050 013 03 INST=0 PIC X(3).
000051 013 03 CODE=2=0 PIC X.
000052 013 03 CODE=5=0 PIC X(5).
000053 013 03 GEO=0 PIC X(4).
000054 013 02 PDD.
000055 013 03 NAME=P PIC X(14).
0000100 00000100
0000200 00000200
0000300 00000300
0000400 00000400
0000500 00000500
0000600 00000600
0000700 00000700
0000800 00000800
0000900 00000900
0001000 00001000
0001100 00001100
0001200 00001200
0001300 00001300
0001400 00001400
0001500 00001500
0001600 00001600
0001700 00001700
0001800 00001800
0001900 00001900
0002000 00002000
0002100 00002100
0002200 00002200
0002300 00002300
0002400 00002400
0002500 00002500
0002600 00002600
0002700 00002700
0002800 00002800
0002900 00002900
0003000 00003000
0003100 00003100
0003200 00003200
0003300 00003300
0003400 00003400
0003500 00003500
0003600 00003600
0003700 00003700
0003800 00003800
0003900 00003900
0004000 00004000
0004100 00004100
0004200 00004200
0004300 00004300
0004400 00004400
0004500 00004500
0004600 00004600
0004700 00004700
0004800 00004800
0004900 00004900
0005000 00005000

```



```

000056 013 U3 INST-P PIC X(13). 00005100
000057 013 U3 CODE-2-P PIC XX. 00005200
000058 013 U3 CODE-5-P PIC X(15). 00005300
000059 013 U3 GEO-P PIC X(4). 00005400
000060 013 U2 DESTINATION-SET. 00005500
000061 013 U3 NAME-U PIC X(14). 00005600
000062 013 U3 INST-U PIC X(13). 00005700
000063 013 U3 CODE-2-D PIC XX. 00005800
000064 013 U3 CODE-5-D PIC X(15). 00005900
000065 013 U3 GEO-D PIC X(4). 00006000
000066 013 *** DATA SCALED BY .001 FOR SMBSMOD. *** 00006100
000067 013 *** MORSAS UNITS ARE UNIT PAX. TONS OR SQ FT *** 00006200
000068 013 *** SMBSMOD UNITS ARE THOUSANDS *** 00006300
000069 013 U2 PAX PIC 999V999. 00006400
000070 013 U2 A-AMMO PIC 999V999. 00006500
000071 013 U2 A-SUP PIC 999V999. 00006600
000072 013 U2 BULK PIC 999V999. 00006700
000073 013 U2 OVER-POL PIC 999V999. 00006800
000074 013 U2 OUT-SZ PIC 999V999. 00006900
000075 013 U2 NAT PIC 999V999. 00007000
000076 013 U2 TOTAL-WT PIC 999V999. 00007100
000077 013 U2 TPSN PIC X(15). 00007200
000078 013 U2 SRC PIC X(11). 00007300
000079 013 U2 FRN PIC X(4). 00007400
000080 013 U2 UTC PIC X(5). 00007500
000081 013 U2 UIC PIC X(6). 00007600
000082 013 U2 AC-NR PIC X(13). 00007700
000083 013 U2 FILLER PIC X(10). 00007800
000084 013 U2 TIME-SAVE. 00007900
000085 013 U3 SAVE-RDD PIC 999. 00008000
000086 013 U3 SAVE-AVAIL PIC 999. 00008100
000087 013 U2 NSDA-SQ-FT PIC 999V999. 00008200
000088 013 U2 FILLER PIC X(12). 00008300
000089 013 U2 PREPO PIC X(13). 00008400
000090 013 U2 OO-NODE PIC 999. 00008500
000091 013 U2 DD-NODE PIC 999. 00008600
000092 013 U2 SVMD PIC X. 00008700
000093 013 U2 FILLER PIC X(35). 00008800
000094 013 FD PRINT-FILE LABEL RECORD OMITTED. 00008900
000095 013 U1 PRINT-LINE PIC X(132). 00009000
000096 013 SD SORT-ING. 00009100
000097 013 U1 SORT-REC. 00009200
000098 013 U2 FILLER PIC X(28). 00009300
000099 013 U2 T-MODE1 PIC X. 00009400
000100 013 U2 FILLER PIC XX. 00009500
000101 013 U2 AVAIL PIC 999. 00009600
000102 013 U2 RDD PIC 999. 00009700
000103 013 U2 FILLER PIC X(23). 00009800
000104 013 U2 CODE-2-D PIC XX. 00009900
000105 013 U2 FILLER PIC X(54). 00010000
000106 013 U2 FILLER PIC XX. 00010100
000107 013 U2 FILLER PIC X(128). 00010200
000108 013 U2 OO-NODE PIC 999. 00010300
000109 013 U2 DD-NODE PIC 999. 00010400
000110 013 U2 FILLER PIC X(36). 00010500
000111 013 FD O-NODE-FILE LABEL RECORD OMITTED. 00010600
000112 013 U1 O-NODE-REC. 00010700

```

```

000113 013 02 ORIG PIC XX.
000114 013 02 FILLER PIC X.
000115 013 02 ORIG-NODE PIC 999.
000116 013 02 FILLER PIC X(74).
000117 013 FD D-NODE-FILE LABEL RECORD OMITTED.
000118 013 01 D-NODE-REC.
000119 013 02 DEST PIC XX.
000120 013 02 FILLER PIC X.
000121 013 02 DEST-NODE PIC 9(3).
000122 013 02 FILLER PIC X(74).
000123 013 COMMON-STORAGE SECTION.
000124 013 77 CARDS-ABC PIC X(84).
000125 013 01 ARRAY-1.
000126 013 03 PKG PIC 9(6).
000127 013 03 RUD PIC 999.
000128 013 03 AVAIL PIC 999.
000129 013 03 DD-NODE PIC 999.
000130 013 03 DD-NODE PIC 999.
000131 013 03 MODE-TS PIC X.
000132 013 03 PAX PIC 9(5)V999.
000133 013 03 NAT PIC 9(5)V999.
000134 013 03 NSDA=NAT PIC 9(5)V999 VALUE ZERO.
000135 013 03 NSDA-SQ-FT PIC 9(5)V999.
000136 013 03 VEH-OUT PIC 9(5)V999 VALUE ZERO.
000137 013 03 POL PIC 9(5)V999.
000138 013 03 FILLER PIC X(29) VALUE SPACES.
000139 013 03 OUT-SZ PIC 9(5)V999.
000140 013 03 NSDA-OVR-C PIC 9(5)V999 VALUE ZERO.
000141 013 03 NSDA-OVR-NC PIC 9(5)V999 VALUE ZERO.
000142 013 03 VEH-OVR-C PIC 9(5)V999 VALUE ZERO.
000143 013 03 VEH-OVR-NC PIC 9(5)V999 VALUE ZERO.
000144 013 03 OVR PIC 9(5)V999.
000145 013 03 NV-OVR-C PIC 9(5)V999 VALUE ZERO.
000146 013 03 VEH-BULK-C PIC 9(5)V999 VALUE ZERO.
000147 013 03 BULK1 PIC 9(5)V999.
000148 013 03 DRY-RESUP PIC 9(5)V999 VALUE ZERO.
000149 013 03 RESUP-AMMO PIC 9(5)V999.
000150 013 03 FILLER PIC X(8) VALUE '999999 '.
000151 013 WORKING-STORAGE SECTION.
000152 013 01 C-CARD.
000153 013 02 FILLER PIC X(8) VALUE ' 5444..'.
000154 013 02 FILLER PIC X(22) VALUE SPACES.
000155 013 02 FILLER PIC XX VALUE 'U..'.
000156 013 02 FILLER PIC X(8) VALUE SPACES.
000157 013 02 FILLER PIC XX VALUE 'U..'.
000158 013 02 FILLER PIC X(38) VALUE SPACES.
000159 013 01 HEADER-1.
000160 013 02 FILLER PIC X(30) VALUE 'ORIGIN
000161 013 02 FILLER PIC X(30) VALUE ' HOU DESTINATION
000162 013 02 FILLER PIC X(30) VALUE ' D AVL PAX A-AMMO A-SUP M KTR'.
000163 013 02 FILLER PIC X(30) VALUE ' K OV/POL OUT NAT TOTAL'.
000164 013 02 FILLER PIC X(12) VALUE ' NSDA-SQFT '.
000165 013 01 LINE-1.
000166 013 02 ORIGIN-SET.
000167 013 03 NAME=0 PIC X(14).
000168 013 03 CODE=2=0 PIC XX.
000169 013 02 FILLER PIC X VALUE SPACE.

```

0001100

••UNCLASSIFIED••

MRG PRE-PROCESSOR PROGRAM MORSAROLL

000170	013	02	LEVEL	PIC X(3).			00011400
000171	013	02	FILLER	PIC X VALUE SPACE.			00011300
000172	013	02	REC-NAME	PIC X(12).			00011400
000173	013	02	FILLER	PIC X VALUE SPACES.			00011500
000174	013	02	RDD	PIC 999.			00011600
000175	013	02	FILLER	PIC X VALUE SPACES.			00011700
000176	013	02	DESTINATION-SET.				00011800
000177	013	03	NAME-D	PIC X(14).			00011900
000178	013	03	CODE-2-D	PIC XX.			00011950
000179	013	02	FILLER	PIC X VALUE SPACE.			00012000
000180	013	02	T-MODE.				00012100
000181	013	03	T-MODE1	PIC X.			00012200
000182	013	02	FILLER	PIC X VALUE SPACES.			00012300
000183	013	02	RT	PIC X.			00012400
000184	013	02	RD	PIC X.			00012500
000185	013	02	FILLER	PIC XXX VALUE SPACE.			00012600
000186	013	02	AVAIL	PIC 999.			00012700
000187	013	02	FILLER	PIC X VALUE SPACE.			00012800
000188	013	02	PAX	PIC 999V999.			00012900
000189	013	02	FILLER	PIC X VALUE SPACES.			00013000
000190	013	02	A-AMMO	PIC 999V999.			
000191	013	02	FILLER	PIC X VALUE SPACES.			
000192	013	02	A-SUP	PIC 999V999.			
000193	013	02	FILLER	PIC X VALUE SPACES.			
000194	013	02	BULK	PIC 999V999.			
000195	013	02	FILLER	PIC X VALUE SPACES.			
000196	013	02	OVER-POL	PIC 999V999.			
000197	013	02	FILLER	PIC X VALUE SPACES.			
000198	013	02	OUT-52	PIC 999V999.			
000199	013	02	FILLER	PIC X VALUE SPACES.			
000200	013	02	NAT	PIC 999V999.			
000201	013	02	FILLER	PIC X VALUE SPACES.			
000202	013	02	TOTAL-WT	PIC 999V999.			
000203	013	02	FILLER	PIC X VALUE SPACES.			
000204	013	02	NSDA-SQ-FT	PIC 999V999.			
000205	013	02	FILLER	PIC X(4) VALUE SPACES.			
000206	013	01	A-CARD.				
000207	013	02	FILLER	PIC X VALUE SPACE.			
000208	013	02	FILLER	PIC X(20) VALUE OTHER SERVICE PKG NR.			
000209	013	02	PKG-NR=0	PIC ZZZZ9.			
000210	013	02	FILLER	PIC X(17) VALUE		RDD= U.	
000211	013	02	RDD	PIC 9(3).			
000212	013	02	FILLER	PIC X(15) VALUE		AVL= U.	
000213	013	02	AVAIL	PIC 9(3).			
000214	013	02	FILLER	PIC XX VALUE			
000215	013	02	NEG	PIC X(14) VALUE SPACES.			
000216	013	01	O-NODE-TABLE.				
000217	013	02	TABLE=0	OCCURS 20 TIMES.			
000218	013	03	ORIGIN	PIC XX.			
000219	013	03	O-NODE	PIC 999.			
000220	013	01	D-NODE-TABLE.				
000221	013	02	TABLE=D	OCCURS 20 TIMES.			
000222	013	03	DESTIN	PIC XX.			
000223	013	03	D-NODE	PIC 999.			
000224	013	01	CARGO-TYPE-TABLE.				
000225	013	02	FILLER	PIC 99 VALUE 1.			
000226	013	02	FILLER	PIC 99 VALUE 15.			

••UNCLASSIFIED••

MRG PRE-PROCESSOR PROGRAM MORSAROLL


```

000227 013 02 FILLER PIC 99 VALUE 12. 00013200
000228 013 02 FILLER PIC 99 VALUE 6. 00013300
000229 013 02 FILLER PIC 99 VALUE 7. 00013400
000230 013 02 FILLER PIC 99 VALUE 2. 00013500
000231 013 02 FILLER PIC 99 VALUE 15. 00013600
000232 013 02 FILLER PIC 99 VALUE 15. 00013700
000233 013 02 FILLER PIC 99 VALUE 4. 00013800
000234 013 02 FILLER PIC 99 VALUE 17. 00013900
000235 013 01 TABLE-CARGO-TYPE REDEFINES CARGO-TYPE-TABLE. 00014100
000236 013 02 CARGO-TYPE PIC 99 OCCURS 10 TIMES. 00014200
000237 013 01 NO-NODE-LINE. 00014300
000238 013 02 FILLER PIC X(19) VALUE '***NODE NOT FOUND *'.
000239 013 02 CODE*2 PIC XX. VALUE SPACES.
000240 016 02 FILLER PIC X. VALUE SPACES.
000241 013 02 FLAG*1 PIC 9 VALUE 2.
000242 016 02 FILLER PIC X. VALUE SPACES.
000243 016 02 TOTAL-WT PIC 999V999.
000244 016 02 BULK PIC 999V999.
000245 016 02 PAX PIC 999V999.
000246 016 02 RDD PIC 999.
000247 016 02 AVAIL PIC 999.
000248 016 02 FILLER PIC X(184).
000249 013 01 COM-PARE.
000250 013 02 T-MODE.
000251 013 03 T-MODE1 PIC X.
000252 013 02 DD-NODE PIC 999.
000253 013 02 DD-NODE PIC 999.
000254 013 02 RDD PIC 999.
000255 013 02 AVAIL PIC 999.
000256 013 01 SAVE-DATA.
000257 013 02 COMPARE-SET.
000258 013 03 T-MODE.
000259 013 04 T-MODE1 PIC X.
000260 013 03 DD-NODE PIC 999.
000261 013 03 DD-NODE PIC 999.
000262 013 03 RDD PIC 999.
000263 013 03 AVAIL PIC 999.
000264 013 02 ADD-SET USAGE IS COMPUTATIONAL.
000265 013 03 PAX PIC 9(5)V999 VALUE ZERO.
000266 013 03 BULK1 PIC 9(5)V999 VALUE ZERO.
000267 013 03 OVR PIC 9(5)V999 VALUE ZERO.
000268 013 03 POL PIC 9(5)V999 VALUE ZERO.
000269 013 03 OUT-52 PIC 9(5)V999 VALUE ZERO.
000270 013 03 NAT PIC 9(5)V999 VALUE ZERO.
000271 013 03 A-AMMU PIC 9(5)V999 VALUE ZERO.
000272 013 03 A-SUP PIC 9(5)V999 VALUE ZERO.
000273 013 03 MSDA-SG-FT PIC 9(5)V999 VALUE ZERO.
000274 013 03 RESUP-AMMU PIC 9(5)V999 VALUE ZERO.
000275 013 02 WTY-TABLE REDEFINES ADD-SET USAGE IS COMP.
000276 013 03 QTY PIC 9(5)V999 OCCURS 10 TIMES.
000277 013 01 DELTA-CARD.
000278 013 02 DELTA-RDD PIC 99.
000279 013 02 DELTA-AVAIL PIC 99.
000280 013 02 FILLER PIC X(76).
000281 013 01 B-CARD.
000282 013 02 FILLER PIC X(14) VALUE SPACES.
000283 013 02 AVAIL PIC 999.

```



```

000284 013 02 FILLER PIC X(15) VALUE ' '.
000285 013 02 DEST-NODE PIC 999.
000286 013 02 FILLER PIC X VALUE SPACES.
000287 013 02 ORIG-NODE PIC 999.
000288 013 02 FILLER PIC X(4) VALUE SPACES.
000289 013 02 T-MODE PIC X VALUE ZERO.
000290 013 02 FILLER PIC X VALUE SPACES.
000291 013 02 PAX PIC 9(15).999 VALUE ZERO.
000292 013 02 FILLER PIC X VALUE SPACES.
000293 013 02 AREG PIC 9(15).999 VALUE ZERO.
000294 013 02 CARGO-TYPE-FBLK PIC ZZZZ9.
000295 013 02 FILLER PIC X(25) VALUE SPACES.
000296 013 77 J PICTURE 99 VALUE 1 USAGE IS COMP.
000297 013 77 PKG-NR PIC 9(15) USAGE IS COMPUTATIONAL VALUE ZERO.
000298 013 77 COMP-POL PIC 9(16) USAGE IS COMP.
000299 013 77 O-CNT PIC 9(4) USAGE IS COMPUTATIONAL VALUE ZERO.
000300 013 77 D-CNT PIC 9(4) USAGE IS COMPUTATIONAL VALUE ZERO.
000301 013 77 LINE-COUNT PIC 99 USAGE IS COMP VALUE 54.
000302 013 77 HALF-DELTA-AVAIL PIC 99 USAGE IS COMP.
000303 013 77 HALF-DELTA-RDD PIC 99 USAGE IS COMP.
000304 017 77 POL-WT PIC 9999999 USAGE IS COMP.
000305 013 PROCEDURE DIVISION.
000306 013 OPEN INPUT MORSAROLL.
000307 013 OPEN INPUT PRINT-FILE PACKAGE-FILE.
000308 013 ***** READ NODE FILES AND SET UP NODE TABLES*****
000309 013 READ-O-NODE.
000310 013 READ O-NODE-FILE AT END GO TO READ-O-NODE.
000311 013 ADD 1 TO O-CNT MOVE ORIG-NODE IN O-NODE-REC
000312 013 TO O-NODE(O-CNT) MOVE ORIG TO ORIGIN(O-CNT).
000313 013 GO TO READ-O-NODE.
000314 013 READ-D-NODE.
000315 013 CLOSE O-NODE-FILE OPEN INPUT D-NODE-FILE.
000316 013 ADD 1 TO D-CNT.
000317 013 READ-DD-NODE.
000318 013 READ D-NODE-FILE AT END GO TO END-D.
000319 013 ADD 1 TO D-CNT MOVE DEST-NODE IN D-NODE-REC
000320 013 TO D-NODE(D-CNT).
000321 013 MOVE DEST TO DESTIN(D-CNT).
000322 013 GO TO READ-DD-NODE.
000323 013 END-D.
000324 013 ADD 1 TO D-CNT.
000325 013 CLOSE D-NODE-FILE.
000326 013 OPEN INPUT CARD-FILE.
000327 013 READ CARD-FILE INTO DELTA-CARD AT END NEXT SENTENCE.
000328 013 COMPUTE HALF-DELTA-AVAIL = DELTA-AVAIL / 2.
000329 013 COMPUTE HALF-DELTA-RDD = DELTA-RDD / 2.
000330 013 A-SORT SECTION.
000331 013 SORT SORT-ING ON ASCENDING KEY RDD AVAIL T-MODE1
000332 013 OO-NODE DD-NODE
000333 013 INPUT PROCEDURE IS SORT-1
000334 013 OUTPUT PROCEDURE IS SMOBSMOD.
000335 013 GO TO END-RUN.
000336 013 SORT-1 SECTION.
000337 013 READ-MORSA.
000338 013 *
000339 013 * ELIMINATE ARMY RECORDS AND INTRA THEATER RECORDS*****
000340 013 *

```

00022700
00022500

000341 013 READ MORSAR-FILE AT END GO TO END-SORT-1.
000342 013 IF SERV IN MORSAR-REC = 'A' GO TO READ-MORSA.
000343 013 *****
000344 013 IF RD IN MORSAR-REC GREATER THAN 45 GO TO READ-MORSA.
000345 013 *****
000346 013 IF T-MODEL IN MORSAR-REC = TO 'A' OR 'P' OR 'S',
000347 013 PERFORM SET-NODES,
000348 013 ELSE GO TO READ-MORSA.
000349 013 IF FLAG-1 NOT LESS THAN 4 PERFORM RULL-ON-DELTA.
000350 013 GO TO READ-MORSA.
000351 013 ROLL-ON-DELTA.
000352 013 *****
000353 013 MOVE RD IN MORSAR-REC TO SAVE-RDU
000354 013 MOVE AVAIL IN MORSAR-REC TO SAVE-AVAIL.
000355 013 DIVIDE DELTA-RDU INTO RD IN MORSAR-REC.
000356 013 DIVIDE DELTA-AVAIL INTO AVAIL IN MORSAR-REC.
000357 013 *****
000358 013 MOVE T-MODEL IN MORSAR-REC TO SVMD.
000359 013 IF SVMD = 'P' MOVE 'U' TO T-MODEL IN MORSAR-REC.
000360 013 IF SVMD = 'S',
000361 013 IF PAX IN MORSAR-REC > 0 MOVE 'J' TO T-MODEL
000362 013 IN MORSAR-REC
000363 013 ELSE MOVE '2' TO T-MODEL IN MORSAR-REC.
000364 013 IF SVMD = 'A',
000365 013 IF NAT IN MORSAR-REC = 0 MOVE '1' TO T-MODEL
000366 013 IN MORSAR-REC
000367 013 ELSE IF PAX IN MORSAR-REC = 0 MOVE '2' TO
000368 013 T-MODEL IN MORSAR-REC
000369 013 ELSE MOVE 'J' TO T-MODEL IN MORSAR-REC.
000370 013 PERFORM CHECK-FOR-PKG-POL
000371 013 ELSE RELEASE SORT-REC FROM MORSAR-REC.
000372 013
000373 013 END-SORT-1.
000374 013 CLOSE MORSAR-FILE.
000375 013 *****
000376 013 SET-NODES SECTION.
000377 013 MOVE 2 TO FLAG-1.
000378 013 O-NODES.
000379 013 IF J = 0-CNT MOVE CODE-2-0 IN MORSAR-REC TO CODE-2
000380 017 MOVE CORR MORSAR-REC TO NO-NODE-LINE
000381 013 MOVE 1 TO FLAG-1 WRITE PRINT-LINE FROM NO-NODE-LINE
000382 013 GO TO J-1.
000383 013 IF CODE-2-0 IN MORSAR-REC = ORIGIN(J) MOVE
000384 013 O-NODE(J) TO 00-NODE IN MORSAR-REC GO TO J-1
000385 013 ELSE ADD 1 TO J GO TO O-NODES.
000386 013 J-1.
000387 013 MOVE 1 TO J.
000388 013 O-NODES.
000389 013 IF J = D-CNT MOVE CODE-2-0 IN MORSAR-REC TO CODE-2
000390 016 MOVE CORR MORSAR-REC TO NO-NODE-LINE
000391 013 MOVE 0 TO FLAG-1 WRITE PRINT-LINE FROM NO-NODE-LINE
000392 013 GO TO SKIP-REC.
000393 013 IF CODE-2-0 IN MORSAR-REC = DESTIN(J) MOVE O-NODE(J)
000394 013 TO 00-NODE IN MORSAR-REC GO TO SKIP-REC
000395 013 ELSE ADD 1 TO J GO TO O-NODES.
000396 013 ***** DO NOT PROCESS RECORD IF NODE NOT FOUND*****
000397 013 SKIP-REC.

00022700
00022500

00022700
00022500

```
000398 013 MOVE 1 TO J.
000399 013 SMOBSMOD SECTION.
000400 013 RETURN SORT-ING RECORD INTO MORSAR-REC AT END GO
000401 013 TO END-S. PERFORM RE-SET-DATES.
000402 013 PERFORM SAVE-INFO.
000403 013 RETURN-AGAIN.
000404 013 RETURN SORT-ING RECORD INTO MORSAR-REC AT END GO
000405 013 TO END-S.
000406 013 RE-SET-DATES.
000407 013 COMPUTE AVAIL IN MORSAR-REC = AVAIL IN MORSAR-REC *
000408 013 DELTA-Avail * HALF-DELTA-Avail.
000409 013 COMPUTE RDD IN MORSAR-REC = RDD IN MORSAR-REC *
000410 013 DELTA-RDD * HALF-DELTA-RDD.
000411 013 P=1.
000412 013 MOVE CORRESPONDING MORSAR-REC TO COM-PARE.
000413 013 IF COM-PARE = COMPARE-SET PERFORM ADD-PROC.
000414 013 ELSE PERFORM SMOBSMOD-PROC THRU END-SMOB.
000415 013 GO TO RETURN-AGAIN.
000416 013 ADD-PROC.
000417 013 PERFORM PACKAGE-DETAIL.
000418 013 ADD CORR MORSAR-REC TO ADD-SET.
000419 013 IF RT IN MORSAR-REC = 'P' MOVE OVER-POL IN MORSAR-REC TO
000420 013 COMP-POL.
000421 013 COMPUTE COMP-POL = COMP-POL * 1000.
000422 013 ADD COMP-POL TO POL IN SAVE-DATA.
000423 013 ELSE ADD OVER-POL IN MORSAR-REC TO OVR IN SAVE-DATA.
000424 013 IF RT IN MORSAR-REC = 'S' AND RD IN MORSAR-REC = 'A'
000425 013 ADD BULK IN MORSAR-REC TO RESUP-AMMO IN SAVE-DATA.
000426 013 ELSE ADD BULK IN MORSAR-REC TO BULK1 IN SAVE-DATA.
000427 013 SMOBSMOD-PROC.
000428 013 * * * A CARD * * *
000429 013 ADD 1 TO PKG-NR MOVE PKG-NR TO PKG-NR-0.
000430 013 MOVE CORR COMPARE-SET TO A-CARD.
000431 013 WRITE PRINT-LINE FROM A-CARD AFTER 2.
000432 013 MOVE A-CARD TO CARDS-ABC.
000433 013 ENTER FORTRAN SMOBS SUBROUTINE.
000434 013 * * * BOTTOM LINE PACKAGE DEATAIL * * *
000435 013 WRITE PACK-REC FROM A-CARD MOVE SPACES TO PACK-REC
000436 013 WRITE PACK-REC ADD 2 TO LINE-COUNT.
000437 013 * * * B CARDS * * *
000438 013 * * * AVAIL * * *
000439 013 MOVE AVAIL IN A-CARD TO AVAIL IN B-CARD.
000440 013 MOVE T-MODEL IN SAVE-DATA TO T-MODEL IN B-CARD.
000441 013 MOVE DD-NODE IN SAVE-DATA TO DD-NODE IN B-CARD.
000442 013 MOVE DD-NODE IN SAVE-DATA TO DD-NODE IN B-CARD.
000443 013 * * * PAX * * *
000444 013 IF PAX IN SAVE-DATA NOT GREATER THAN ZERO GO TO CARGO.
000445 013 MOVE PAX IN SAVE-DATA TO PAX IN B-CARD. MOVE 1 TO
000446 013 CARGO-TYPE-FBLK. WRITE PRINT-LINE FROM B-CARD.
000447 013 MOVE B-CARD TO CARDS-ABC.
000448 013 ENTER FORTRAN SMOBS SUBROUTINE.
000449 013 MOVE ZERO TO PAX IN B-CARD.
000450 013 * * * CARGO * * *
000451 013 CARGO.
000452 013 PERFORM A-REG VARYING J FROM 2 BY 1 UNTIL J = 11.
000453 013 MOVE ZERO TO AREG IN B-CARD.
000454 013 * * * C-CARD * * *
```



```

000455 013 MOVE C-CARD TO CARDS-ABC.
000456 013 ENTER FORTRAN SMOBS SUBROUTINE.
000457 013 * * * WRITE ARRAY FOR PATTY. * *
000458 013 MOVE CORR ADD-SET TO ARRAY-1 MOVE CORR COMPARE-SET
000459 013 TO ARRAY-1. MOVE PKG-MR TO PKG.
000460 013 MOVE T-MODE IN B-CARD TO MODE-TS.
000461 013 ENTER FORTRAN SMOBS2 SUBROUTINE.
000462 013 MOVE 1 TO J.
000463 013 SAVE-INFO.
000464 013 PERFORM PACKAGE-DETAIL.
000465 013 MOVE CORR MORSAR-REC TO COMPARE-SET.
000466 013 MOVE CORR MORSAR-REC TO ADD-SET.
000467 013 IF RT IN MORSAR-REC = 'S' AND MD IN MORSAR-REC = 'A'
000468 013 MOVE BULK IN MORSAR-REC TO RESUP-AMMO IN SAVE-DATA
000469 013 MOVE ZERO TO BULK1 IN SAVE-DATA
000470 013 ELSE MOVE BULK IN MORSAR-REC TO BULK1 IN SAVE-DATA
000471 013 MOVE ZERO TO RESUP-AMMO IN SAVE-DATA.
000472 013 IF RT IN MORSAR-REC = 'P' MOVE OVER-POL IN
000473 013 MORSAR-REC TO COMP-POL COMPUTE COMP-POL =
000474 013 COMP-POL * 1000 MOVE COMP-POL TO POL IN SAVE-DATA
000475 013 MOVE ZERO TO OVR IN SAVE-DATA
000476 013 ELSE MOVE OVER-POL IN MORSAR-REC TO OVR IN SAVE-DATA
000477 013 MOVE ZERO TO POL IN SAVE-DATA.
000478 013 END-SMOB.
000479 013 EXIT.
000480 013 A-REG.
000481 013 IF QTY(J) GREATER THAN ZERO, MOVE QTY(J) TO
000482 013 AREG IN B-CARD, MOVE CARGO-TYPE(J) TO
000483 013 CARGO-TYPE-FBLK, MOVE B-CARD TO CARDS-ABC,
000484 013 ENTER FORTRAN SMOBS SUBROUTINE,
000485 013 WRITE PRINT-LINE FROM B-CARD.
000486 013 END-S.
000487 013 PERFORM SMOBSMOD-PROC THRU CARGO.
000488 013 MOVE C-CARD TO CARDS-ABC.
000489 013 ENTER FORTRAN SMOBS1 SUBROUTINE.
000490 013 END-RUN SECTION.
000491 013 CLOSE PRINT-FILE PACKAGE-FILE.
000492 013 CLOSE CARD-FILE.
000493 013 STOP RUN.
000494 013 * * * EACH RECORD MAKING UP A PACKAGE IS PRINTED * *
000495 013 PACKAGE-DETAIL SECTION.
000496 013 CHECK-COUNT.
000497 013 IF LINE-COUNT LESS THAN 54 GO TO PRINT-DETAIL.
000498 013 SET-HEADER.
000499 013 MOVE ZERO TO LINE-COUNT WRITE PACK-REC FROM
000500 013 HEADER-1 AFTER ADVANCING TOP-OF-PAGE.
000501 013 MOVE SPACES TO PACK-REC WRITE PACK-REC.
000502 013 PRINT-DETAIL.
000503 013 MOVE CORR MORSAR-REC TO LINE-1.
000504 013 MOVE SMD TO T-MODE1 IN LINE-1.
000505 013 MOVE SAVE-RDU TO RDU IN LINE-1.
000506 013 MOVE SAVE-AVAIL TO AVAIL IN LINE-1.
000507 013 WRITE PACK-REC FROM LINE-1.
000508 013 ADD 1 TO LINE-COUNT.
000509 013 SEPARATE-POL SECTION.
000510 013 CHECK-FOR-PKG-POL.
000511 013 IF BULK IN MORSAR-REC > 0 PERFORM MOVE-2 THRU MOVE-1

```

00029000

00029000

00029020

00029050

00029000

00029000

00029000

00029000

00030000

00030000

00030000

00030000

00030000


```

000512 013      ELSE PERFORM MOVE-1.
000513 013      MOVE-2.
000514 013      * * * MAKE SEPARATE RECORD FOR PACKAGE POL. * *
000515 013      MOVE OVER-POL IN MORSAR-REC TO COMP-POL.
000516 013      MOVE ZERO TO OVER-POL IN MORSAR-REC.
000517 013      * * * 1000/7.213 BBL5 PER ST = 138.639 ST PER 1000 BBL5. * *
000518 013      COMPUTE POL-WT = 138.639 * COMP-POL.
000519 014      SUBTRACT POL-WT FROM TOTAL-WT IN MORSAR-REC.
000520 013      RELEASE SORT-REC FROM MORSAR-REC.
000521 013      MOVE POL-WT TO TOTAL-WT IN MORSAR-REC MOVE COMP-POL TO
000522 013      OVER-POL IN MORSAR-REC.
000523 015      MOVE ZERO TO BULK IN MORSAR-REC.
000524 013      * * * MAKE RECORD FOR POL. SET NODE = 191 FOR GALVESTON. * *
000525 013      MOVE-1.
000526 013      MOVE 191 TO 00-NODE IN MORSAR-REC.
000527 013      RELEASE SORT-REC FROM MORSAR-REC.

```

END ELT.

```

@ELT,L 24PRG,SMOBS
ELT007 RL1870 08/16-07:40:57-(12.)
000001 000      SUBROUTINE SMOBS
000002 002      COMMON DUMMY, CARDS(14),A1(16),A2(16)
000003 000      102      FORMAT (14A6)
000004 000      WRITE (2,102) CARDS
000005 000      RETURN
000006 002      ENTRY SMOBS2
000007 002      WRITE (3,103) A1
000008 002      WRITE (3,103) A2
000009 002      103      FORMAT (16A6)
000010 002      RETURN
000011 000      ENTRY SMOBS1
000012 000      WRITE (2,102) CARDS
000013 000      WRITE (2,102) CARDS
000014 000      ENDFILE 2
000015 002      ENDFILE 3
000016 000      RETURN
000017 000      END

```

END ELT.

MRG PRE-PROCESSOR PROGRAM TUCAROLL

MRG PRE-PROCESSOR PROGRAM TUCAROLL

MRG PRE-PROCESSOR PROGRAM TUCAROLL

MRG PRE-PROCESSOR PROGRAM TUCAROLL

••UNCLASSIFIED••

MRG PRE-PROCESSOR PROGRAM TUCAROLL

000056	029	05 TTYPE	PICTURE IS 9(01)•	
000057	029	88 FTYPE	PICTURE IS 9(01)	VALUE IS 1.
000058	029	88 F2TYPE	PICTURE IS 9(01)	VALUE IS 2.
000059	029	88 F3TYPE	PICTURE IS 9(01)	VALUE IS 3.
000060	029	05 F3IDNO	PICTURE IS 9(03)•	
000061	029	05 ULC	PICTURE IS X(01)•	
000062	029	05 DEPLOYMENT	PICTURE IS X(01)•	
000063	029	05 SERVICE	PICTURE IS X(01)•	
000064	029	05 SECURE	PICTURE IS X(01)•	
000065	029	05 SHORT-NAME	PICTURE IS X(15)•	
000066	029	05 VALIDITY	PICTURE IS X(01)•	
000067	029	88 TYPEA-RCRD	PICTURE IS X(01)	VALUE IS 'T'.
000068	029	05 TYPE-STATUS	PICTURE IS X(01)•	
000069	029	88 CANCELLED	PICTURE IS X(01)	VALUE IS 'C'.
000070	029	05 F1CLASS	PICTURE IS X(01)•	
000071	029	03 F1DATA.		
000072	029	05 TYPE-NAME	PICTURE IS X(54)•	
000073	029	05 UIC	PICTURE IS X(06)•	
000074	029	05 CREATE-DATE	PICTURE IS 9(06)•	
000075	029	05 CHANGE-DATE	PICTURE IS 9(06)•	
000076	029	05 AUTH-WAR-STR	PICTURE IS 9(05)•	
000077	029	05 NOPAX	PICTURE IS 9(05)•	
000078	029	05 F2REQ	PICTURE IS 9(03)•	
000079	029	05 NOF2	PICTURE IS 9(03)•	
000080	029	05 NEWF1	PICTURE IS X(01)•	
000081	029	02 F1-SRC.		
000082	029	05 SRC	PICTURE IS X(12)•	
000083	033	05 FILLER	PICTURE IS X(01)•	
000084	029	01 F2-REC.		
000085	029	03 TUF2.		
000086	029	05 F2RKEY	PICTURE IS X(12)•	
000087	029	05 F2CLASS	PICTURE IS X(01)•	
000088	029	05 NEWF2	PICTURE IS X(01)•	
000089	029	05 HLC	PICTURE IS X(01)•	
000090	029	05 FILLER	PICTURE IS X(03)•	
000091	029	05 F2SQFT	PICTURE IS 9(06)•	
000092	029	05 F2STON	PICTURE IS 9(06)•	
000093	029	05 F2MTON	PICTURE IS 9(06)•	
000094	029	05 HBPOL	PICTURE IS 9(06)•	
000095	029	05 F3REQ	PICTURE IS 9(03)•	
000096	029	05 NOF3	PICTURE IS 9(03)•	
000097	029	03 F2-FIL.		
000098	029	05 FILLER	PICTURE IS X(90)•	
000099	029	01 F3-REC.		
000100	029	03 TUF3.		
000101	029	05 F3RKEY	PICTURE IS X(12)•	
000102	029	05 F3CLASS	PICTURE IS X(01)•	
000103	029	05 NEWF3	PICTURE IS X(01)•	
000104	029	05 F3DESC	PICTURE IS X(14)•	
000105	029	05 LENGTH	PICTURE IS 9(04)•	
000106	029	05 WIDTH	PICTURE IS 9(03)•	
000107	029	05 HEIGHT	PICTURE IS 9(03)•	
000108	029	05 F3SQFT	PICTURE IS 9(04)•	
000109	029	05 PIECES	PICTURE IS 9(03)•	
000110	029	05 F3STON	PICTURE IS 9(06)•	
000111	029	05 F3MTON	PICTURE IS 9(06)•	
000112	029	05 FILLER	PICTURE IS X(03)•	

••UNCLASSIFIED••

MRG PRE-PROCESSOR PROGRAM TUCAROLL

MRG PRE-PROCESSOR PROGRAM TUCAROLL

MRG PRE-PROCESSOR PROGRAM TUCAROLL

MRG PRE-PROCESSOR PROGRAM TUCAROLL

MRG PRE-PROCESSOR PROGRAM TUCAROLL

000113 029 03 F3-FIL. PICTURE IS X(78).
000114 029 05 FILLER DATA RECORD IS LOG-REC.
000115 029 SD SORT-FILE
000116 029 01 LOG-REC. PICTURE IS 9(05).
000117 029 05 SEGNO
000118 029 05 COTYPE. PICTURE IS A(01).
000119 029 07 COTYPE1 PICTURE IS 9(01).
000120 029 07 COTYPE2 PICTURE IS 9(05).
000121 029 05 PERTO
000122 029 05 CRCAT. PICTURE IS X(01).
000123 029 07 CRCAT1 PICTURE IS X(01).
000124 029 07 CRCAT2 PICTURE IS X(01).
000125 029 05 CONCO PICTURE IS X(01).
000126 029 05 SWFT PICTURE IS 9(06).
000127 029 05 STOF PICTURE IS 9(06).
000128 029 05 STON PICTURE IS 9(06).
000129 029 05 MTON. PICTURE IS 9(01).
000130 029 07 MTOF PICTURE IS 9(06).
000131 029 07 WHMTO PICTURE IS 9(01).
000132 029 07 TEMET PICTURE IS 9(01).
000133 029 05 HELCO PICTURE IS A(01).
000134 029 05 SRC. PICTURE IS X(09).
000135 029 07 SRC19 PICTURE IS X(01).
000136 029 07 SRC10 PICTURE IS X(02).
000137 029 07 SRC11-12 PICTURE IS X(05).
000138 029 05 UTC PICTURE IS X(06).
000139 029 05 UIC PICTURE IS X(05).
000140 029 05 FILLER PICTURE IS X(05).
000141 029 66 LEVCO RENAMES CRCAT THRU CONCO.
000142 029 66 UNSTR RENAMES SWFT.
000143 029 66 UNAME RENAMES STOF THRU TEMET.
000144 029 66 UPIND RENAMES HELCO.
000145 029 FD F5
000146 029 LABEL RECORDS ARE STANDARD
000147 029 DATA RECORD IS RF5.
000148 029 01 RF5 PICTURE IS X(53/61).
000149 029 COMMON-STORAGE SECTION.
000150 029 01 LINE1.
000151 029 05 SRC1.
000152 029 07 SRC1-9
000153 029 07 SRCPAK
000154 029 05 LINE-1
000155 029 05 FILLER
000156 029 05 ULC
000157 029 05 FILLER
000158 029 05 UIC
000159 029 05 STR
000160 029 05 STON-A0
000161 029 05 STON-B0
000162 029 05 STON-B1
000163 029 05 STON-A1
000164 029 05 STON-J1
000165 029 05 STON-B2C
000166 029 05 STON-B2D
000167 029 05 STON-A2C
000168 029 05 STON-A2D
000169 029 05 STON-J2D

MRG PRE-PROCESSOR PROGRAM TUCAROLL

MRG PRE-PROCESSOR PROGRAM TUCAROLL

000170	029	05 STON-J2C	PIC Z(5)9.
000171	029	05 STON-A3C	PIC Z(5)9.
000172	029	05 STON-J3C	PIC Z(5)9.
000173	029	05 STON-J7C	PIC Z(5)9.
000174	029	05 STON-M7C	PIC Z(5)9.
000175	029	05 STON-UE	PIC Z(5)9.
000176	029	05 STON-AS	PIC Z(5)9.
000177	029	01 LINE2.	
000178	029	05 SRC2	PIC X(11).
000179	029	05 LINE-2	PIC X(11) VALUE *2.
000180	029	05 NAME	PIC X(12).
000181	029	05 PAX	PIC Z(5)9.
000182	029	05 SQFT-A0	PIC Z(5)9.
000183	029	05 SQFT-B0	PIC Z(5)9.
000184	029	05 SQFT-B1	PIC Z(5)9.
000185	029	05 SQFT-A1	PIC Z(5)9.
000186	029	05 FILLER	PIC X(6) VALUE SPACES.
000187	029	05 SQFT-B2C	PIC Z(5)9.
000188	029	05 SQFT-B2D	PIC Z(5)9.
000189	029	05 SQFT-A2C	PIC Z(5)9.
000190	029	05 SQFT-A2D	PIC Z(5)9.
000191	029	05 FILLER	PIC X(12) VALUE SPACES.
000192	029	05 SQFT-A3C	PIC Z(5)9.
000193	029	05 FILLER	PIC X(18) VALUE SPACES.
000194	029	05 SQFT-NSDA	PIC Z(5)9.
000195	029	05 SQFT-VEH	PIC Z(5)9.
000196	029	WORKING-STORAGE SECTION.	
000197	029	77 C-FLAG	PICTURE IS X VALUE SPACE.
000198	029	77 E-FLAG	PICTURE IS X VALUE SPACE.
000199	029	77 COUNT	PICTURE IS 9(06).
000200	029	77 TUNEXT	PICTURE IS 9.
000201	029	77 N2REQ	PICTURE IS 9(10) COMPUTATIONAL.
000202	029	77 N3REQ	PICTURE IS 9(10) COMPUTATIONAL.
000203	029	77 FSKEY	PICTURE IS 9(06).
000204	029	77 UNCOUNT	PICTURE IS 9(06) VALUE 18.
000205	029	77 CAT	PICTURE IS 9(02).
000206	029	01 CAT-ARRAYS.	
000207	029	02 STON-A OCCURS 18 TIMES	PIC 9(06).
000208	029	02 SQFTA OCCURS 18 TIMES	PIC 9(06).
000209	029	01 HEAD-LINE.	
000210	029	05 WD01-05	PIC X(30) VALUE * SRC
000211	029	05 WD06-10	PIC X(30) VALUE * AU B0
000212	029	05 WD11-14	PIC X(24) VALUE * B2C B2D A2C A2D.
000213	029	05 WD15-18	PIC X(24) VALUE * J2D J2C A3C J3C.
000214	029	05 WD19-22	PIC X(24) VALUE * J7C AMMO UE/B AS/A.
000215	029	01 F5-WORK.	
000216	029	05 F5SEQ	PICTURE IS 9(05).
000217	029	05 FILLER	PICTURE IS X(61).
000218	029	01 F5-DESC.	
000219	029	05 F5MAKCRDS	PIC 9(10) VALUE 3500 COMPUTATIONAL.
000220	029	05 F5RECLEN	PIC 9(10) VALUE 66 COMPUTATIONAL.
000221	029	05 F5FILLER	PIC 9(10) VALUE U COMPUTATIONAL.
000222	029	05 F5KEYLEN	PIC 9(10) VALUE 6 COMPUTATIONAL.
000223	029	05 F5NR0FA	PIC 9(10) VALUE 5 COMPUTATIONAL.
000224	029	05 F5NI0FA	PIC 9(10) VALUE 20 COMPUTATIONAL.
000225	029	05 F5WD011.	
000226	029	07 F5WD011A	PIC X(03) VALUE SPACES.

000227	029	07 F5WD11B	PIC X(03)	VALUE SPACES.
000228	029	01 F5-INFO.		
000229	029	05 F5RBLKS	PICTURE IS 9(10)	COMPUTATIONAL.
000230	029	05 F5IBLKS	PICTURE IS 9(10)	COMPUTATIONAL.
000231	029	05 F5N10FB	PICTURE IS 9(10)	COMPUTATIONAL.
000232	029	05 F5NRGRD	PICTURE IS 9(10)	COMPUTATIONAL.
000233	029	05 F5NR10F	PICTURE IS 9(10)	COMPUTATIONAL.
000234	029	05 F5NDEL	PICTURE IS 9(10)	COMPUTATIONAL.
000235	029	05 F5NREAD	PICTURE IS 9(10)	COMPUTATIONAL.
000236	029	05 F5NF10F	PICTURE IS 9(10)	COMPUTATIONAL.
000237	029	05 F5NWRITE	PICTURE IS 9(10)	COMPUTATIONAL.
000238	029	01 F5-INFORMATION.		
000239	029	05 F5RBLKS	PICTURE IS 9(10).	
000240	029	05 FILLER	PICTURE IS X(02)	VALUE IS SPACES.
000241	029	05 F5IBLKS	PICTURE IS 9(10).	
000242	029	05 FILLER	PICTURE IS X(02)	VALUE IS SPACES.
000243	029	05 F5N10FB	PICTURE IS 9(10).	
000244	029	05 FILLER	PICTURE IS X(02)	VALUE IS SPACES.
000245	029	05 F5NRGRD	PICTURE IS 9(10).	
000246	029	05 FILLER	PICTURE IS X(02)	VALUE IS SPACES.
000247	029	05 F5NR10F	PICTURE IS 9(10).	
000248	029	05 FILLER	PICTURE IS X(02)	VALUE IS SPACES.
000249	029	05 F5NDEL	PICTURE IS 9(10).	
000250	029	05 FILLER	PICTURE IS X(02)	VALUE IS SPACES.
000251	029	05 F5NREAD	PICTURE IS 9(10).	
000252	029	05 FILLER	PICTURE IS X(02)	VALUE IS SPACES.
000253	029	05 F5NF10F	PICTURE IS 9(10).	
000254	029	05 FILLER	PICTURE IS X(02)	VALUE IS SPACES.
000255	029	05 F5NWRITE	PICTURE IS 9(10).	
000256	029	05 FILLER	PICTURE IS X(02)	VALUE IS SPACES.
000257	029	01 CARD-WORK.		
000258	029	05 CUT-OFF	PICTURE IS 9(05).	
000259	029	05 TCOPI-FLAG	PICTURE IS X(05).	
000260	029	88 TCOPI	PICTURE IS X(05)	VALUE IS 'TCOPY'.
000261	029	05 LOADS-FLAG	PICTURE IS X(05).	
000262	029	88 LOADS	PICTURE IS X(05)	VALUE IS 'LOADS'.
000263	029	05 PRINT-FLAG1	PICTURE IS X(05).	
000264	029	88 PRINT-TY	PICTURE IS X(05)	VALUE IS 'TYPEA'.
000265	029	05 PRINT-FLAG2	PICTURE IS X(05).	
000266	029	88 PRINT-F1	PICTURE IS X(05)	VALUE IS 'ABF11'.
000267	029	05 PRINT-FLAG3	PICTURE IS X(05).	
000268	029	88 PRINT-L1	PICTURE IS X(05)	VALUE IS 'ABF10'.
000269	029	05 PRINT-FLAG4	PICTURE IS X(05).	
000270	029	88 PRINT-F2	PICTURE IS X(05)	VALUE IS 'TUF21'.
000271	029	05 PRINT-FLAG5	PICTURE IS X(05).	
000272	029	88 PRINT-L2	PICTURE IS X(05)	VALUE IS 'TUF20'.
000273	029	05 PRINT-FLAG6	PICTURE IS X(05).	
000274	029	88 PRINT-F3	PICTURE IS X(05)	VALUE IS 'TUF31'.
000275	029	05 PRINT-FLAG7	PICTURE IS X(05).	
000276	029	88 PRINT-L0	PICTURE IS X(05)	VALUE IS 'FIF20'.
000277	029	05 FILLER	PICTURE IS X(30).	
000278	029	01 ERROR-WORK.		
000279	029	05 NUMB	PICTURE IS X(06).	
000280	029	05 FILLER	PICTURE IS X(01)	VALUE IS SPACE.
000281	029	05 MESSAGE		
000282	029	PROCEDURE DIVISION.		
000283	029	DECLARATIVES.	PICTURE IS X(125).	

••UNCLASSIFIED••

MRG PRE-PROCESSOR PROGRAM TUCMAROLL

```

000284 029 BAD-TAPE SECTION. USE AFTER STANDARD ERROR PROCEDURE ON TUCHA.
000285 029 TUCHA-ERROR-PROC.
000286 029 DISPLAY 'TAPE ERROR ON TUCHA'.
000287 029 DISPLAY 'TYPE LOST - PREVIOUS LOG REC'.
000288 029 MOVE TUNEXT TO NUMB.
000289 029 MOVE LOG-REC TO MESSAGE.
000290 029 DISPLAY ERROR=WORK.
000291 029 MOVE 'E' TO E-FLAG.
000292 029 IF TUNEXT EQUALS 1 MOVE 'C' TO C-FLAG.
000293 029 END DECLARATIVES.
000294 029 NORMAL-PROCESSING-SECTION.
000295 029 OPEN-FILES.
000296 029 OPEN INPUT TUCHA CARD-FILE OUTPUT PRINT-FILE.
000297 029 IF TCOPI OPEN OUTPUT SAVE-FILE.
000298 029 IF LOADS OPEN OUTPUT F5.
000299 029 READ-CARD.
000300 029 READ CARD-FILE INTO CARD=WORK AT END GO TO END-CARD.
000301 029 READ CARD-FILE INTO CARD=WORK AT END GO TO END-CARD.
000302 029 END-CARD.
000303 029 CLOSE CARD-FILE.
000304 029 MOVE ZERO TO COUNT.
000305 029 MOVE ZERO TO FSKEY.
000306 029 SORT SORT-FILE ON ASCENDING KEY SRC IN LOG-REC
000307 029 UIC IN LOG-REC CDTYPE CREAT CONCO IN LOG-REC
000308 029 INPUT PROCEDURE IS NEXT-TUCHA THRU END-TUCHA
000309 029 OUTPUT PROCEDURE IS LOAD-F5 THRU END-LOAD.
000310 029 GO TO CLOSE-FILES.
000311 029 NEXT-TUCHA.
000312 029 IF COUNT IS NOT LESS THAN CUT-OFF GO TO CLOSE-TUCHA.
000313 029 READ TUCHA AT END GO TO CLOSE-TUCHA.
000314 029 IF E-FLAG EQUALS 'E' MOVE SPACE TO E-FLAG GO TO NEXT-TUCHA.
000315 029 IF TCOPI WRITE SAVE-REC FROM FI-REC.
000316 029 IF TUTYPE EQUALS 1 IF TYPEA=KCRD GO TO USE-TYPEA
000317 029 ELSE GO TO USE-ABF1
000318 029 ELSE IF F2TYPE GO TO USE-TUF2
000319 029 ELSE IF F3TYPE GO TO USE-TUF3
000320 029 ELSE GO TO TYPE-ERR.
000321 029 TYPE-ERR.
000322 029 MOVE TUTYPE TO NUMB.
000323 029 MOVE 'TYPE TUCHA RECORD NOT EXPECTED' TO MESSAGE.
000324 029 WRITE PRINT-LINE FROM ERROR=WORK.
000325 029 GO TO CLOSE-TUCHA.
000326 029 USE-TYPEA.
000327 029 IF PRINT-TY MOVE TYPEA TO PRINT-LINE PERFORM PRINT-OUT.
000328 029 GO TO NEXT-TUCHA.
000329 029 USE-ABF1.
000330 029 IF PRINT-FI
000331 029 MOVE SRC IN FI-REC TO PRINT-LINE PERFORM PRINT-OUT
000332 029 MOVE ABF1 TO PRINT-LINE PERFORM PRINT-OUT.
000333 029 IF CANCELLED MOVE 'C' TO C-FLAG GO TO NEXT-TUCHA
000334 029 ELSE MOVE SPACE TO C-FLAG.
000335 029 ADD 1 TO COUNT.
000336 029 MOVE SPACES TO LOG-REC.
000337 029 MOVE COUNT TO SEQNO IN LOG-REC.
000338 029 MOVE 'FI' TO CDTYPE IN LOG-REC.
000339 029 MOVE NOPAX TO PERTO IN LOG-REC.
000340 029 MOVE ULC IN FI-REC TO LEVCO.

```

••UNCLASSIFIED••

MRG PRE-PROCESSOR PROGRAM TUCMAROLL

•UNCLASSIFIED•

MRG PRE-PROCESSOR PROGRAM TUCAROLL

```

000341 029 MOVE AUTH-WAR-STR TO UNSTR.
000342 029 MOVE SHORT-NAME TO UNAME.
000343 029 MOVE DEPLOYMENT TO DPIND.
000344 029 MOVE SRC IN FI-REC TO SRC IN LOG-REC.
000345 029 MOVE UTC IN ABFI TO UTC IN LOG-REC.
000346 029 MOVE UIC IN ABFI TO UIC IN LOG-REC.
000347 029 IF PRINT-LI MOVE LOG-REC TO PRINT-LINE PERFORM PRINT-OUT.
000348 029 IF SRCID EQUALS '1' RELEASE LOG-REC.
000349 029 MOVE NOF2 TO NZREQ.
000350 029 IF NZREQ EQUALS ZERO MOVE 1 TO TUNEXT
000351 029 ELSE MOVE 2 TO TUNEXT.
000352 029 GO TO NEXT-TUCHA.
000353 029
000354 029 USE-TUF2.
000355 029 IF PRINT-F2 MOVE TUF2 TO PRINT-LINE PERFORM PRINT-OUT.
000356 029 IF C-FLAG EQUALS 'C' GO TO NEXT-TUCHA.
000357 029 ADD 1 TO COUNT.
000358 029 MOVE COUNT TO SEGNO IN LOG-REC.
000359 029 MOVE 'F2' TO CDTYPE IN LOG-REC.
000360 029 MOVE ZEROS TO PERTO IN LOG-REC.
000361 029 MOVE CARGO TO CRCAT IN LOG-REC.
000362 029 MOVE CONTAINER TO CONCO IN LOG-REC.
000363 029 MOVE F2SWFT TO SWFT IN LOG-REC.
000364 029 MOVE ZERO TO STOF IN LOG-REC.
000365 029 MOVE F2STON TO STON IN LOG-REC.
000366 029 MOVE ZERO TO MTUF IN LOG-REC.
000367 029 IF BULKPOL MOVE MBPOL TO WMTO IN LOG-REC.
000368 029 ELSE MOVE F2MTON TO WMTO IN LOG-REC.
000369 029 MOVE ZERO TO TEMET IN LOG-REC.
000370 029 MOVE HLC IN TUF2 TO HELCO IN LOG-REC.
000371 029 IF PRINT-L2 MOVE LOG-REC TO PRINT-LINE PERFORM PRINT-OUT.
000372 029 MOVE UTC IN ABFI TO UTC IN LOG-REC.
000373 029 IF SRCID EQUALS '1' RELEASE LOG-REC.
000374 029 SUBTRACT 1 FROM NZREQ.
000375 029 IF NZREQ EQUALS ZERO MOVE 1 TO TUNEXT
000376 029 ELSE MOVE NOF3 TO NZREQ
000377 029 IF NZREQ EQUALS ZERO MOVE 2 TO TUNEXT
000378 029 ELSE MOVE 3 TO TUNEXT.
000379 029 GO TO NEXT-TUCHA.
000380 029
000381 029 USE-TUF3.
000382 029 IF PRINT-F3 MOVE TUF3 TO PRINT-LINE PERFORM PRINT-OUT.
000383 029 SUBTRACT 1 FROM NZREQ.
000384 029 IF NZREQ EQUALS ZERO MOVE 2 TO TUNEXT.
000385 029 GO TO NEXT-TUCHA.
000386 029
000387 029 CLOSE-TUCHA.
000388 029 CLOSE TUCHA.
000389 029 IF TCOY CLOSE SAVE-FILE.
000390 029
000391 029 END-TUCHA.
000392 029
000393 029 LOAD-F5.
000394 029 RETURN SORT-FILE RECORD INTO F5-WORK AT END GO TO LAST-SUM.
000395 029 ADD 1 TO F5KEY.
000396 029 MOVE F5KEY TO F5SEQ IN F5-WORK.
000397 029 IF PRINT-FO MOVE F5-WORK TO PRINT-LINE PERFORM PRINT-OUT.
000398 029 IF LOAD5 WRITE RF5 FROM F5-WORK INVALID KEY GO TO KEY-ERR.
000399 029 IF CDTYPE EQUALS 'F1' GO TO FIROLL ELSE GO TO F2ROLL.
000400 029 F2ROLL.
000401 029 IF CRCAT1 EQUALS 'A' OR 'C' GO TO VEHICLE.
000402 032

```

•UNCLASSIFIED•

MRG PRE-PROCESSOR PROGRAM TUCAROLL

UNCLASSIFIED

MRG PRE-PROCESSOR PROGRAM TUCHAROLL

```
000400 IF CRCAT1 EQUALS 'B' GO TO NSDA.
000401 IF CRCAT1 EQUALS 'J' GO TO NON-VEH.
000402 IF CRCAT1 EQUALS 'M' GO TO AMMO.
000403 MOVE CRCAT TO NUMB.
000404 MOVE 'CRCAT NOT EXPECTED' TO MESSAGE.
000405 WRITE PRINT-LINE FROM ERROR-WORK.
000406 GO TO LOAD-F5.
000407 VEHICLE.
000408 IF CONCO EQUALS 'M' OR 'C' MOVE 8 TO CAT
000409 ELSE MOVE 9 TO CAT.
000410 IF CRCAT2 EQUALS 'O' MOVE 1 TO CAT.
000411 IF CRCAT2 EQUALS 'I' MOVE 4 TO CAT.
000412 IF CRCAT2 EQUALS '3' MOVE 12 TO CAT.
000413 ADD SWFT TO SWFTAIL7.
000414 GO TO SUM-CAT.
000415 NSDA.
000416 IF CONCO EQUALS 'C' OR 'B' MOVE 6 TO CAT
000417 ELSE MOVE 7 TO CAT.
000418 IF CRCAT2 EQUALS 'O' MOVE 2 TO CAT.
000419 IF CRCAT2 EQUALS 'I' MOVE 3 TO CAT.
000420 ADD SWFT TO SWFTAIL6.
000421 GO TO SUM-CAT.
000422 NON-VEH.
000423 IF CONCO EQUALS 'C' OR 'B' MOVE 11 TO CAT
000424 ELSE MOVE 10 TO CAT.
000425 IF CRCAT2 EQUALS 'O' MOVE 1 TO CAT.
000426 IF CRCAT2 EQUALS 'I' MOVE 5 TO CAT.
000427 IF CRCAT2 EQUALS '3' MOVE 13 TO CAT.
000428 IF CRCAT2 EQUALS '7' MOVE 14 TO CAT.
000429 GO TO SUM-CAT.
000430 AMMO.
000431 MOVE 15 TO CAT.
000432 SUM-CAT.
000433 ADD STON TO STONALCAT1.
000434 ADD SWFT TO SWFTAILCAT1.
000435 IF CAT 15 LESS THAN 14 ADD STON TO STONAI16.
000436 ADD STON TO STONAI17.
000437 ELSE
000438 GO TO LOAD-F5.
000439 FIROLL.
000440 IF F5SEQ EQUALS 1 GO TO FISAVE.
000441 PRINT-SUM.
000442 IF UNCOUNT EQUALS 18 MOVE ZERO TO UNCOUNT
000443 WRITE PRINT-LINE FROM HEAD-LINE
000444 AFTER ADVANCING TOP-OF-PAGE LINES.
000445 ADD 1 TO UNCOUNT.
000446 MOVE STONAI11 TO STON-AD IN LINE1.
000447 MOVE STONAI12 TO STON-B0 IN LINE1.
000448 MOVE STONAI13 TO STON-B1 IN LINE1.
000449 MOVE STONAI14 TO STON-A1 IN LINE1.
000450 MOVE STONAI15 TO STON-J1 IN LINE1.
000451 MOVE STONAI16 TO STON-B2C IN LINE1.
000452 MOVE STONAI17 TO STON-B2D IN LINE1.
000453 MOVE STONAI18 TO STON-A2C IN LINE1.
000454 MOVE STONAI19 TO STON-A2D IN LINE1.
000455 MOVE STONAI10 TO STON-J2D IN LINE1.
000456 MOVE STONAI11 TO STON-J2C IN LINE1.
000457 MOVE STONAI12 TO STON-A3C IN LINE1.
```

UNCLASSIFIED

```

000455 029 MOVE STONA(13) TO STON-J3C IN LINE1.
000456 029 MOVE STONA(14) TO STON-J7C IN LINE1.
000457 029 MOVE STONA(15) TO STON-M7C IN LINE1.
000458 029 MOVE STONA(16) TO STON-UE IN LINE1.
000459 029 MOVE STONA(17) TO STON-AS IN LINE1.
000460 029 WRITE PRINT-LINE FROM LINE1 AFTER ADVANCING 2 LINES.
000461 029 MOVE SQFTA(1) TO SQFT-AD IN LINE2.
000462 029 MOVE SQFTA(2) TO SQFT-BD IN LINE2.
000463 029 MOVE SQFTA(3) TO SQFT-B1 IN LINE2.
000464 029 MOVE SQFTA(4) TO SQFT-A1 IN LINE2.
000465 029 MOVE SQFTA(6) TO SQFT-B2C IN LINE2.
000466 029 MOVE SQFTA(7) TO SQFT-B2D IN LINE2.
000467 029 MOVE SQFTA(8) TO SQFT-A2C IN LINE2.
000468 029 MOVE SQFTA(9) TO SQFT-A2D IN LINE2.
000469 029 MOVE SQFTA(12) TO SQFT-A3C IN LINE2.
000470 029 MOVE SQFTA(16) TO SQFT-NSDA.
000471 029 MOVE SQFTA(17) TO SQFT-VEH IN LINE2.
000472 029 WRITE PRINT-LINE FROM LINE2.
000473 029 ENTER FORTRAN SQFTU SUBROUTINE.
000474 029 FSAVE.
000475 029 MOVE SRC19 IN LOG-REC TO SRC1-9 IN LINE1.
000476 029 MOVE SRC11-12 TO SRCPAR IN LINE1.
000477 029 MOVE LEVCO TO ULC IN LINE1.
000478 029 MOVE UTC IN LOG-REC TO UTC IN LINE1.
000479 029 MOVE UNSTR TO STR IN LINE1.
000480 029 MOVE SRC1 IN LINE1 TO SRC2 IN LINE2.
000481 029 MOVE UNAME TO NAME IN LINE2.
000482 029 MOVE PERTO IN LOG-REC TO PAX IN LINE2.
000483 029 PERFORM ZERO-ARRAYS VARYING CAT FROM 1 BY 1
000484 029 UNTIL CAT IS GREATER THAN 18.
000485 029 GO TO LOAD-F5.
000486 029 ZERO-ARRAYS.
000487 029 MOVE ZEROS TO STONA(CAT) SQFTALCAT).
000488 029 LAST-SUM.
000489 029 PERFORM PRINT-SUM.
000490 029 ENTER FORTRAN EOF12 SUBROUTINE.
000491 029 END-LOAD.
000492 029 EXIT.
000493 029 KEY-ERR.
000494 029 MOVE F5KEY TO NUMB.
000495 029 MOVE 'INVALID KEY FOR FIF2 FILE' TO MESSAGE.
000496 029 MOVE ERROR-WORK TO PRINT-LINE.
000497 029 PRINT-OUT.
000498 029 WRITE PRINT-LINE.
000499 029 CLOSE-FILES.
000500 029 IF LOAD5 CLOSE F5 USING F5-INFO.
000501 029 MOVE 'F5 INFORMATION AREA' TO PRINT-LINE.
000502 029 PERFORM PRINT-OUT.
000503 029 MOVE CORRESPONDING F5-INFO TO F5-INFORMATION.
000504 029 MOVE F5-INFORMATION TO PRINT-LINE.
000505 029 PERFORM PRINT-OUT.
000506 029 CLOSE PRINT-FILE.
000507 029 STOP RUN.

```

END ELT.

WELT.L 32MARTIN.SOFTU
ELT007 RL1870 08/16-07:41:01-11.)
000001 000 SUBROUTINE SOFTU
000002 000 COMMON DUMMY, LINE1(22), LINE2(22)
000003 000 WRITE (12,4) LINE1
000004 000 WRITE (12,4) LINE2
000005 000 RETURN
000006 000 4 FORMAT (22A6)
000007 000 ENTRY EOF12
000008 000 ENDFILE 12
000009 000 RETURN
000010 000 END

END ELT.

BHGG MRG PRE-PROCESSOR PROGRAM PUMCUSROLL ••UNCLASSIFIED•• L.O

WELT.L 32MARTIN.POMSUM

ELT007 RL1870 08/16-07:41:03-119.)

000001	015	IDENTIFICATION DIVISION.
000002	015	PROGRAM-ID. LOAD-FIF2.
000003	015	AUTHOR. G.L.MARTIN
000004	015	DATE-WRITTEN. JANUARY 1975
000005	015	DATE-COMPILED.
000006	015	INSTALLATION. US ARMY CAA.
000007	015	SECURITY. UNCLASSIFIED.
000008	015	REMARKS. PROGRAM LOADS CAMP FIF2 FILE FROM A
000009	015	NONSTANDARD POMCUS FILE WITH SRC APPENDED TO THE
000010	015	TUFI RECORD.
000011	015	ENVIRONMENT DIVISION.
000012	015	CONFIGURATION SECTION.
000013	015	SOURCE-COMPUTER. UNIVAC-1108.
000014	015	OBJECT-COMPUTER. UNIVAC-1108.
000015	015	INPUT-OUTPUT SECTION.
000016	015	FILE-CONTROL.
000017	015	SELECT PRINT-FILE ASSIGN TO PRINTER.
000018	015	SELECT CARD-FILE ASSIGN TO CARD-READER.
000019	015	SELECT POMCUS ASSIGN TO UNISERVO POMCUS.
000020	015	SELECT SAVE-FILE ASSIGN TO UNISERVO SAVE.
000021	015	SELECT SORT-FILE ASSIGN TO MASS-STORAGE AA.
000022	015	SELECT F5 ASSIGN TO MASS-STORAGE 32FIF2
000023	015	ORGANIZATION IS INDEXED
000024	015	FILE-DESCRIPTION IS F5-DESC
000025	015	ACCESS MODE IS RANDOM
000026	015	SYMBOLIC KEY IS F5KEY.
000027	015	DATA DIVISION.
000028	015	FILE SECTION.
000029	015	FD PRINT-FILE
000030	015	LABEL RECORDS ARE OMITTED
000031	015	DATA RECORD IS PRINT-LINE.
000032	015	01 PRINT-LINE PICTURE IS X(132).
000033	015	FD CARD-FILE
000034	015	LABEL RECORDS ARE OMITTED
000035	015	DATA RECORD IS INPUT-CARD.
000036	015	01 INPUT-CARD PICTURE IS X(80).
000037	015	FD SAVE-FILE
000038	015	LABEL RECORDS ARE OMITTED
000039	015	BLOCK CONTAINS 24 RECORDS
000040	015	RECORD CONTAINS 138 CHARACTERS
000041	015	DATA RECORD IS SAVE-REC.
000042	015	01 SAVE-REC PICTURE IS X(138).
000043	015	FD POMCUS
000044	015	LABEL RECORDS ARE OMITTED
000045	015	RECORDING MODE IS 1
000046	015	BLOCK CONTAINS 10 RECORDS
000047	015	RECORD CONTAINS 120 CHARACTERS
000048	015	DATA RECORDS ARE F1-REC, F2-REC, F3-REC.
000049	015	01 F1-REC.
000050	015	02 TUFI.
000051	015	03 TYPEA.
000052	015	05 FILLER
000053	015	05 TUTYPE
000054	015	88 F1TYPE
000055	015	88 F2TYPE

MRG PRE-PROCESSOR PROGRAM POMCUSROLL

MRG PRE-PROCESSOR PROGRAM POMCUSROLL

MRG

000056 015 88 F3TYPE PICTURE IS 9(01) VALUE IS J.

000057 015 05 FILLER PICTURE IS X(01).

000058 015 05 UTC PICTURE IS X(05).

000059 015 05 POMCUS=CODE1 PICTURE IS X(01).

000060 015 05 FILLER PICTURE IS X(04).

000061 015 05 NOPAX PICTURE IS 9(05).

000062 015 05 F2REQ PICTURE IS 9(03).

000063 015 05 FILLER PICTURE IS X(44).

000064 015 05 UTC PICTURE IS X(06).

000065 015 05 FILLER PICTURE IS X(13).

000066 015 05 SRC PICTURE IS X(12).

000067 015 05 FILLER PICTURE IS X(19).

000068 015 66 AUTH=WAR-STR RENAMES NOPAX.

000069 015 66 NOF2 RENAMES F2REQ.

000070 015 01 F2-REC.

000071 015 03 TUF2.

000072 015 05 FILLER PICTURE IS X(06).

000073 015 05 TUTYPE2 PICTURE IS X(01).

000074 015 05 FILLER PICTURE IS X(01).

000075 015 05 UTC PICTURE IS X(05).

000076 015 05 POMCUS=CODE2 PICTURE IS A(01).

000077 015 88 TAT PICTURE IS A(01) VALUE IS 'A'.

000078 015 88 OTHER PICTURE IS A(01) VALUE IS 'B'.

000079 015 88 PREPO PICTURE IS A(01) VALUE IS 'C'.

000080 015 05 FILLER PICTURE IS X(19).

000081 015 05 CARGO.

000082 015 07 CARGO1 PICTURE IS A(01).

000083 015 88 BULKPOL PICTURE IS A(01) VALUE IS 'G'.

000084 015 07 CARGO2 PICTURE IS 9(01).

000085 015 05 CONTAINER PICTURE IS A(01).

000086 015 05 F2SOFT PICTURE IS 9(06).

000087 015 05 FILLER PICTURE IS X(02).

000088 015 05 F2STON PICTURE IS 9(06).

000089 015 05 FILLER PICTURE IS X(01).

000090 015 05 F2MTON PICTURE IS 9(06).

000091 015 05 FILLER PICTURE IS X(01).

000092 015 05 HLC PICTURE IS A(01).

000093 015 05 F3REQ PICTURE IS 9(03).

000094 015 05 FILLER PICTURE IS X(08).

000095 015 05 F2UIC PICTURE IS X(06).

000096 015 05 FILLER PICTURE IS X(13).

000097 015 05 F2SRC PICTURE IS X(12).

000098 015 05 FILLER PICTURE IS X(19).

000099 015 66 NOF3 RENAMES F3REQ.

000100 015 01 F3-REC.

000101 015 03 TUF3.

000102 015 05 FILLER PICTURE IS X(06).

000103 015 05 TUTYPE3 PICTURE IS 9(01).

000104 015 05 FILLER PICTURE IS X(01).

000105 015 05 UTC3 PICTURE IS X(05).

000106 015 05 POMCUS=CODE3 PICTURE IS A(01).

000107 015 05 FILLER PICTURE IS X(04).

000108 015 05 CARGO3 PICTURE IS X(03).

000109 015 05 F3DNO PICTURE IS 9(03).

000110 015 05 F3DESC PICTURE IS X(13).

000111 015 05 LENGTH PICTURE IS 9(04).

000112 015 05 WIDTH PICTURE IS 9(03).

MRG PRE-PROCESSOR PROGRAM POMCUSROLL

MRG PRE-PROCESSOR PROGRAM POMCUSROLL

MRG

MRG PRE-PROCESSOR PROGRAM POMCUSROLL

MRG PRE-PROCESSOR PROGRAM POMCUSROLL

MRG

```

000113 015 05 HEIGHT PICTURE IS 9(U3).
000114 015 05 F3SFT PICTURE IS 9(U4).
000115 015 05 PIECES PICTURE IS 9(U3).
000116 015 05 F3STON PICTURE IS 9(U6).
000117 015 05 F3MTON PICTURE IS 9(U6).
000118 015 05 FILLER PICTURE IS 9(U4).
000119 015 05 F3UIC PICTURE IS 9(U6).
000120 015 05 FILLER PICTURE IS 9(U3).
000121 015 05 F3SRC PICTURE IS 9(U2).
000122 015 05 FILLER PICTURE IS 9(U3).
000123 016 05 SORT-FILE DATA RECORD IS SORT-REC.
000124 016 01 SORT-REC.
000125 016 05 FILLER PICTURE IS 9(U5).
000126 017 05 SRTYPE PICTURE IS 9(U2).
000127 016 05 FILLER PICTURE IS 9(U5).
000128 016 05 SRCAT PICTURE IS 9(U3).
000129 016 05 FILLER PICTURE IS 9(U4).
000130 016 05 SRUIC PICTURE IS 9(U6).
000131 015 05 FILLER PICTURE IS 9(U1).
000132 016 05 SRPON PICTURE IS 9(U1).
000133 015 05 FILLER PICTURE IS 9(U3).
000134 015 FD F5
000135 015 LABEL RECORDS ARE STANDARD
000136 015 DATA RECORD IS RF5.
000137 015 01 RF5 PICTURE IS 9(53/6).
000138 015 COMMON-STORAGE SECTION.
000139 015 01 LINE1.
000140 015 05 SRC1.
000141 015 07 SRC1-9
000142 015 07 SRCPAR
000143 015 05 LINE-1
000144 015 05 FILLER
000145 015 05 ULC
000146 015 05 FILLER
000147 015 05 UTC
000148 015 05 STR
000149 015 05 STON-AQ
000150 015 05 STON-BQ
000151 015 05 STON-B1
000152 015 05 STON-A1
000153 015 05 STON-J1
000154 015 05 STON-B2C
000155 015 05 STON-B2D
000156 015 05 STON-A2C
000157 015 05 STON-A2D
000158 015 05 STON-J2C
000159 015 05 STON-J2C
000160 015 05 STON-A3C
000161 015 05 STON-J3C
000162 015 05 STON-J7C
000163 015 05 STON-M7C
000164 015 05 STON-UE
000165 015 05 STON-AS
000166 015 01 LINE2.
000167 015 05 UIC
000168 015 05 FILLER
000169 015 05 LINE-2

```

000170	015	05 NAME	PIC X(12).
000171	015	05 PAX	PIC Z(5)9.
000172	015	05 SQFT-A0	PIC Z(5)9.
000173	015	05 SQFT-B0	PIC Z(5)9.
000174	015	05 SQFT-B1	PIC Z(5)9.
000175	015	05 SQFT-A1	PIC Z(5)9.
000176	015	05 FILLER	PIC X(6) VALUE SPACES.
000177	015	05 SQFT-B2C	PIC Z(5)9.
000178	015	05 SQFT-B2D	PIC Z(5)9.
000179	015	05 SQFT-A2C	PIC Z(5)9.
000180	015	05 SQFT-A2D	PIC Z(5)9.
000181	015	05 FILLER	PIC X(12) VALUE SPACES.
000182	015	05 SQFT-A3C	PIC Z(5)9.
000183	015	05 FILLER	PIC X(18) VALUE SPACES.
000184	015	05 SQFT-NSDA	PIC Z(5)9.
000185	015	05 SQFT-VEH	PIC Z(5)9.
000186	015	WORKING-STORAGE SECTION.	
000187	015	77 C-FLAG	PICTURE IS X VALUE SPACE.
000188	015	77 E-FLAG	PICTURE IS X VALUE SPACE.
000189	015	77 COUNT	PICTURE IS 9(06).
000190	015	77 TUNEXT	PICTURE IS 9.
000191	015	77 N2REQ	PICTURE IS 9(10) COMPUTATIONAL.
000192	015	77 N3REQ	PICTURE IS 9(10) COMPUTATIONAL.
000193	015	77 FSKEY	PICTURE IS 9(06).
000194	015	77 UNCOUNT	PICTURE IS 9(06) VALUE 10.
000195	015	77 PT	PICTURE IS 9(01) VALUE 1.
000196	015	77 CAT	PICTURE IS 9(02).
000197	015	77 UIC-SEQ	PICTURE IS 9(02).
000198	015	77 NO-UICS	PICTURE IS 9(02) VALUE ZERO.
000199	015	01 UIC-ARRAY.	
000200	015	03 UIC-TABLE OCCURS 100 TIMES.	
000201	015	05 OLD-UIC	PICTURE IS X(06).
000202	015	05 NEW-UIC	PICTURE IS X(06).
000203	015	01 UIC-CARD.	
000204	015	03 NOT-UIC	PICTURE IS X(06).
000205	015	03 FILLER	PICTURE IS X(01).
000206	015	03 POM-UIC	PICTURE IS X(06).
000207	015	01 CAT-ARRAYS.	
000208	015	03 POMCUS-TYPE OCCURS 2 TIMES.	
000209	015	05 STONA OCCURS 18 TIMES PIC 9(06).	
000210	015	05 SQFTA OCCURS 18 TIMES PIC 9(06).	
000211	016	01 LOG-REC.	
000212	016	05 SEQNO	PICTURE IS 9(05).
000213	016	05 COTYPE.	
000214	016	07 COTYPE1	PICTURE IS A(01).
000215	016	07 COTYPE2	PICTURE IS 9(01).
000216	016	05 PERTO	PICTURE IS 9(05).
000217	016	05 CRCAT.	
000218	016	07 CRCAT1	PICTURE IS X(01).
000219	016	07 CRCAT2	PICTURE IS X(01).
000220	016	05 CONCO	PICTURE IS X(01).
000221	016	05 SQFT	PICTURE IS 9(06).
000222	016	05 STOF	PICTURE IS 9(02).
000223	016	05 STON	PICTURE IS 9(06).
000224	016	05 MTON.	
000225	016	07 MTOF	PICTURE IS 9(01).
000226	016	07 WHMT0	PICTURE IS 9(06).


```
000227 016 07 TEMET PICTURE IS 9(01).
000228 016 05 HELCO PICTURE IS A(01).
000229 016 05 SRC.
000230 016 07 SRC19 PICTURE IS X(09).
000231 016 07 SRC10 PICTURE IS X(01).
000232 016 07 SRC11-12 PICTURE IS X(02).
000233 016 05 UTC PICTURE IS X(05).
000234 016 05 UIC PICTURE IS X(06).
000235 016 05 FILLER PICTURE IS X(01).
000236 016 05 POMCUS-CODE PICTURE IS A(01).
000237 016 05 FILLER PICTURE IS X(03).
000238 016 66 LEVCO RENAMES CRCAT THRU CONCU.
000239 016 66 UNSTR RENAMES SWFT.
000240 016 66 UNAME RENAMES STOF THRU TEMET.
000241 016 66 OPIND RENAMES HELCO.
000242 015 01 HEAD-LINE.
000243 015 05 W001-05 PIC X(30) VALUE , SRC ULC B1 A1 J1.
000244 015 05 W006-10 PIC X(30) VALUE , AU B0 B20 A2C A20.
000245 015 05 W011-14 PIC X(24) VALUE , B2C J2C A3C J3C.
000246 015 05 W015-18 PIC X(24) VALUE , J2V J2C A3C J3C.
000247 015 05 W019-22 PIC X(24) VALUE , J7C AMMO UE/B AS/A.
000248 015 01 F5-WORK.
000249 015 05 F5SEQ PICTURE IS 9(05).
000250 015 05 FILLER PICTURE IS X(61).
000251 015 01 F5-DESC.
000252 015 05 F5MAXRCROS PIC 9(10) VALUE 3500 COMPUTATIONAL.
000253 015 05 F5RECLEN PIC 9(10) VALUE 66 COMPUTATIONAL.
000254 015 05 FILLER PIC 9(10) VALUE 0 COMPUTATIONAL.
000255 015 05 F5KEYLEN PIC 9(10) VALUE 6 COMPUTATIONAL.
000256 015 05 F5NROFA PIC 9(10) VALUE 5 COMPUTATIONAL.
000257 015 05 F5NIOFA PIC 9(10) VALUE 20 COMPUTATIONAL.
000258 015 05 F5W011.
000259 015 07 F5W011A PIC X(03) VALUE SPACES.
000260 015 07 F5W011B PIC X(03) VALUE SPACES.
000261 015 01 F5-INFO.
000262 015 05 F5RBLKS PICTURE IS 9(10) COMPUTATIONAL.
000263 015 05 F51BLKS PICTURE IS 9(10) COMPUTATIONAL.
000264 015 05 F5NIOFB PICTURE IS 9(10) COMPUTATIONAL.
000265 015 05 F5NRCRD PICTURE IS 9(10) COMPUTATIONAL.
000266 015 05 F5NRIOF PICTURE IS 9(10) COMPUTATIONAL.
000267 015 05 F5NDEL PICTURE IS 9(10) COMPUTATIONAL.
000268 015 05 F5NREAD PICTURE IS 9(10) COMPUTATIONAL.
000269 015 05 F5NRIOF PICTURE IS 9(10) COMPUTATIONAL.
000270 015 05 F5NWRITE PICTURE IS 9(10) COMPUTATIONAL.
000271 015 01 F5-INFORMATION.
000272 015 05 F5RBLKS PICTURE IS 9(10).
000273 015 05 FILLER PICTURE IS X(02) VALUE IS SPACES.
000274 015 05 F51BLKS PICTURE IS 9(10).
000275 015 05 FILLER PICTURE IS X(02) VALUE IS SPACES.
000276 015 05 F5NIOFB PICTURE IS 9(10).
000277 015 05 FILLER PICTURE IS X(02) VALUE IS SPACES.
000278 015 05 F5NRCRD PICTURE IS 9(10).
000279 015 05 FILLER PICTURE IS X(02) VALUE IS SPACES.
000280 015 05 F5NRIOF PICTURE IS 9(10).
000281 015 05 FILLER PICTURE IS X(02) VALUE IS SPACES.
000282 015 05 F5NDEL PICTURE IS 9(10).
000283 015 05 FILLER PICTURE IS X(02) VALUE IS SPACES.
```



```

000284 015 05 F5MREAD PICTURE IS 9(10)• VALUE IS SPACES•
000285 015 05 FILLER PICTURE IS X(02)
000286 015 05 F5MFI0F PICTURE IS 9(10)• VALUE IS SPACES•
000287 015 05 FILLER PICTURE IS X(02)
000288 015 05 F5MWRITE PICTURE IS 9(10)• VALUE IS SPACES•
000289 015 05 FILLER PICTURE IS X(02)
000290 015 01 CARD-WORK•
000291 015 05 CUT-OFF PICTURE IS 9(05)•
000292 015 05 TCOPY-FLAG PICTURE IS X(05)•
000293 015 88 TCOPY PICTURE IS X(05)• VALUE IS 'TCOPY'•
000294 015 05 LOAD5-FLAG PICTURE IS X(05)•
000295 015 88 LOAD5 PICTURE IS X(05)• VALUE IS 'LOAD5'•
000296 015 05 PRINT-FLAG1 PICTURE IS X(05)•
000297 015 88 PRINT-TY PICTURE IS X(05)• VALUE IS 'TYPEA'•
000298 015 05 PRINT-FLAG2 PICTURE IS X(05)•
000299 015 88 PRINT-F1 PICTURE IS X(05)• VALUE IS 'TUF11'•
000300 015 05 PRINT-FLAG3 PICTURE IS X(05)•
000301 015 88 PRINT-L1 PICTURE IS X(05)• VALUE IS 'TUF10'•
000302 015 05 PRINT-FLAG4 PICTURE IS X(05)•
000303 015 88 PRINT-F2 PICTURE IS X(05)• VALUE IS 'TUF21'•
000304 015 05 PRINT-FLAG5 PICTURE IS X(05)•
000305 015 88 PRINT-L2 PICTURE IS X(05)• VALUE IS 'TUF20'•
000306 015 05 PRINT-FLAG6 PICTURE IS X(05)•
000307 015 88 PRINT-F3 PICTURE IS X(05)• VALUE IS 'TUF31'•
000308 015 05 PRINT-FLAG7 PICTURE IS X(05)•
000309 015 88 PRINT-L0 PICTURE IS X(05)• VALUE IS 'FIF20'•
000310 015 05 FILLER PICTURE IS X(30)•
000311 015 01 ERROR-WORK•
000312 015 05 NUMB PICTURE IS X(06)•
000313 015 05 FILLER PICTURE IS X(01)
000314 015 05 MESSAGE PICTURE IS X(125)•
000315 015 05 PROCEDURE DIVISION•
000316 015 05 OPEN-FILES•
000317 015 05 OPEN INPUT POMCUS CARD-FILE OUTPUT PRINT-FILE•
000318 015 05 IF TCOPY OPEN OUTPUT SAVE-FILE•
000319 015 05 IF LOAD5 OPEN OUTPUT F5•
000320 015 05 READ-CARD•
000321 015 05 READ CARD-FILE INTO CARD-WORK AT END GO TO END-CARD•
000322 015 05 READ CARD-FILE INTO CARD-WORK AT END GO TO END-CARD•
000323 015 05 MOVE CARD-WORK TO PRINT-LINE PERFORM PRINT-OUT•
000324 015 05 NEXT-CARD•
000325 015 05 ADD 1 TO NO-UICS•
000326 015 05 READ CARD-FILE INTO UIC-CARD AT END GO TO END-CARD•
000327 015 05 MOVE UIC-CARD TO PRINT-LINE PERFORM PRINT-OUT•
000328 015 05 MOVE POM-UIC TO OLD-UIC(NO-UICS)•
000329 015 05 MOVE NOT-UIC TO NEW-UIC(NO-UICS)•
000330 015 05 GO TO NEXT-CARD•
000331 015 05 END-CARD•
000332 015 05 CLOSE CARD-FILE•
000333 015 05 MOVE ZERO TO COUNT•
000334 015 05 MOVE ZERO TO F5KEY•
000335 016 05 SORT-CONTROL•
000336 016 05 SORT SORT-FILE ON ASCENDING KEY SRUIC SRPOM SRTYPE SRCAT
000337 015 05 INPUT PROCEDURE IS NEXT-POMCUS THRU END-POMCUS
000338 015 05 OUTPUT PROCEDURE IS LOAD-F5 THRU END-LOAD•
000339 015 05 GO TO CLOSE-FILES•
000340 015 05 NEXT-POMCUS•

```

UNCLASSIFIED

MRG PRE-PROCESSOR PROGRAM POMCUSROLL

```
000341 015 IF COUNT IS NOT LESS THAN CUF-OFF GO TO CLOSE-POMCUS.
000342 015 READ POMCUS AT END GO TO CLOSE-POMCUS.
000343 015 IF E-FLAG EQUALS 'E' MOVE SPACE TO E-FLAG GO TO NEXT-POMCUS.
000344 015 IF TCOPY WRITE SAVE-REC FROM FI-REC.
000345 015 IF F1TYPE GO TO USE-TUF1
000346 015 ELSE IF F2TYPE GO TO USE-TUF2
000347 015 ELSE IF F3TYPE GO TO USE-TUF3
000348 015 ELSE GO TO TYPE-ERR.
000349 015 TYPE-ERR.
000350 015 MOVE TUTYPE TO NUMB.
000351 015 MOVE 'TYPE POMCUS RECORD NOT EXPECTED' TO MESSAGE.
000352 015 WRITE PRINT-LINE FROM ERROR-WORK.
000353 015 GO TO CLOSE-POMCUS.
000354 015 USE-TUF1.
000355 015 IF PRINT-F1
000356 015 MOVE SRC IN FI-REC TO PRINT-LINE PERFORM PRINT-OUT
000357 015 MOVE TUF1 TO PRINT-LINE PERFORM PRINT-OUT.
000358 015 ADD 1 TO COUNT.
000359 015 MOVE SPACES TO LOG-REC.
000360 015 MOVE COUNT TO SEQNO IN LOG-REC.
000361 015 MOVE 'FI' TO CDTYPE IN LOG-REC.
000362 015 MOVE NOPAX TO PERTO IN LOG-REC.
000363 015 MOVE AUTH-WAR-STR TO UNSTR.
000364 015 MOVE SRC IN FI-REC TO SRC IN LOG-REC.
000365 015 MOVE ZEROS TO SRC11-12.
000366 015 MOVE UTC IN TUF1 TO UTC IN LOG-REC.
000367 015 MOVE UIC IN TUF1 TO UIC IN LOG-REC.
000368 015 MOVE POMCUS-CODE1 TO POMCUS-CODE.
000369 015 IF PRINT-L1 MOVE LOG-REC TO PRINT-LINE PERFORM PRINT-OUT.
000370 016 RELEASE SORT-REC FROM LOG-REC.
000371 015 PERFORM UIC-SEARCH VARYING UIC-SEQ FROM 1 BY 1
000372 015 UNTIL UIC-SEQ EQUALS NO-UICS.
000373 015 MOVE NOF2 TO NZREG.
000374 015 IF NZREQ EQUALS ZERO MOVE 1 TO TUNEXT
000375 015 ELSE MOVE 2 TO TUNEXT.
000376 015 GO TO NEXT-POMCUS.
000377 015 USE-TUF2.
000378 015 IF PRINT-F2 MOVE TUF2 TO PRINT-LINE PERFORM PRINT-OUT.
000379 015 ADD 1 TO COUNT.
000380 015 MOVE COUNT TO SEQNO IN LOG-REC.
000381 015 MOVE 'F2' TO CDTYPE IN LOG-REC.
000382 015 MOVE ZEROS TO PERTO IN LOG-REC.
000383 015 MOVE CARGO TO CRCAT IN LOG-REC.
000384 015 MOVE CONTAINER TO CONCO IN LOG-REC.
000385 015 MOVE F2SQFT TO SQFT IN LOG-REC.
000386 015 MOVE ZERO TO STOF IN LOG-REC.
000387 015 MOVE F2STON TO STON IN LOG-REC.
000388 015 MOVE F2MTON TO WMTON IN LOG-REC.
000389 015 MOVE ZERO TO TEMET IN LOG-REC.
000390 015 MOVE HLC IN TUF2 TO HELCU IN LOG-REC.
000391 015 MOVE UTC IN TUF1 TO UTC IN LOG-REC.
000392 018 MOVE F2UIC TO UIC IN LOG-REC.
000393 015 MOVE POMCUS-CODE2 TO POMCUS-CODE.
000394 015 IF PRINT-L2 MOVE LOG-REC TO PRINT-LINE PERFORM PRINT-OUT.
000395 016 RELEASE SORT-REC FROM LOG-REC.
000396 015 PERFORM UIC-SEARCH VARYING UIC-SEQ FROM 1 BY 1
000397 015 UNTIL UIC-SEQ EQUALS NO-UICS.
```

UNCLASSIFIED

MRG PRE-PROCESSOR PROGRAM POMCUSROLL

```

000396 015 SUBTRACT 1 FROM N2REG.
000399 015 IF N2REG EQUALS ZERO MOVE 1 TO TUNEXT
000400 015 ELSE MOVE N0F3 TO N3REG
000401 015 IF N3REG EQUALS ZERO MOVE 2 TO TUNEXT
000402 015 ELSE MOVE 3 TO TUNEXT.
000403 015 GO TO NEXT-POMCUS.
000404 015 USE-TUF3.
000405 015 IF PRINT-F3 MOVE TUF3 TO PRINT-LINE PERFORM PRINT-OUT.
000406 015 SUBTRACT 1 FROM N3REG.
000407 015 IF N3REG EQUALS ZERO MOVE 2 TO TUNEXT.
000408 015 GO TO NEXT-POMCUS.
000409 015 CLOSE-POMCUS.
000410 015 CLOSE POMCUS.
000411 015 IF 1COPY CLOSE SAVE-FILE.
000412 015 END-POMCUS.
000413 015 EXIT.
000414 015 UIC-SEARCH.
000415 015 IF UIC IN LOG-REC EQUAL OLD-UIC(UIC-SEQ)
000416 016 MOVE NEW-UIC(UIC-SEQ) TO UIC IN LOG-REC
000417 019 RELEASE SORT-REC FROM LOG-REC
000418 019 MOVE OLD-UIC(UIC-SEQ) TO UIC IN LOG-REC.
000419 015 LOAD-F5.
000420 015 RETURN SORT-FILE RECORD INTO F5-WORK AT END GO TO LAST-SUM.
000421 017 MOVE F5-WORK TO LOG-REC.
000422 015 ADD 1 TO F5KEY.
000423 015 MOVE F5KEY TO F5SEQ IN F5-WORK.
000424 015 IF PRINT-LO MOVE F5-WORK TO PRINT-LINE PERFORM PRINT-OUT.
000425 015 IF LOAD5 WRITE F5-WORK FROM F5-WORK INVALID KEY GO TO KEY-ERR.
000426 015 IF COTYPE EQUALS 'F1' GO TO F1ROLL ELSE GO TO F2ROLL.
000427 015 F2ROLL.
000428 015 IF PT EQUALS 1 AND POMCUS-CODE EQUALS 'C'
000429 015 PERFORM PRINT-SUM MOVE 2 TO PT.
000430 015 IF CRCAT1 EQUALS 'A' OR 'C' GO TO VEHICLE.
000431 015 IF CRCAT1 EQUALS 'B' GO TO NSDA.
000432 015 IF CRCAT1 EQUALS 'J' GO TO NON-VEH.
000433 015 IF CRCAT1 EQUALS 'M' GO TO AMMO.
000434 015 MOVE CRCAT TO NUMB.
000435 015 MOVE 'CRCAT NOT EXPECTED' TO MESSAGE.
000436 015 WRITE PRINT-LINE FROM ERROR-WORK.
000437 015 GO TO LOAD-F5.
000438 015 VEHICLE.
000439 015 IF CONCO EQUALS 'H' OR 'C' MOVE 8 TO CAT
000440 015 ELSE MOVE 9 TO CAT.
000441 015 IF CRCAT2 EQUALS '0' MOVE 1 TO CAT.
000442 015 IF CRCAT2 EQUALS '1' MOVE 4 TO CAT.
000443 015 IF CRCAT2 EQUALS '3' MOVE 12 TO CAT.
000444 015 ADD SUFT TO SUFTA(PT, 17).
000445 015 GO TO SUM-CAT.
000446 015 NSDA.
000447 015 IF CONCO EQUALS 'C' OR 'B' MOVE 6 TO CAT
000448 015 ELSE MOVE 7 TO CAT.
000449 015 IF CRCAT2 EQUALS '0' MOVE 2 TO CAT.
000450 015 IF CRCAT2 EQUALS '1' MOVE 3 TO CAT.
000451 015 ADD SUFT TO SUFTA(PT, 16).
000452 015 GO TO SUM-CAT.
000453 015 NON-VEH.
000454 015 IF CONCO EQUALS 'C' OR 'B' MOVE 11 TO CAT

```


••UNCLASSIFIED••

MRG PRE-PROCESSOR PROGRAM POMCUSHOLL

```

000455 015 ELSE MOVE 10 TO CAT.
000456 015 IF CRCAT2 EQUALS '0' MOVE 1 TO CAT.
000457 015 IF CRCAT2 EQUALS '1' MOVE 5 TO CAT.
000458 015 IF CRCAT2 EQUALS '3' MOVE 13 TO CAT.
000459 015 IF CRCAT2 EQUALS '7' MOVE 14 TO CAT.
000460 015 GO TO SUM-CAT.

000461 015 AMMO.
000462 015 MOVE 15 TO CAT.
000463 015 SUM-CAT.
000464 015 ADD STON TO STONA(PT, CAT).
000465 015 ADD SFTT TO SFTTA(PT, CAT).
000466 015 IF CAT IS LESS THAN 14 ADD STON TO STONA(PT, 16)
000467 015 ELSE ADD STON TO STONA(PT, 17).
000468 015 GO TO LOAD-F5.
000469 015 FIROLL.
000470 015 IF FSSEQ EQUALS 1 GO TO F1SAVE.
000471 015 PRINT-SUM.
000472 015 IF UNCOUNT EQUALS 10 MOVE ZERO TO UNCOUNT
000473 015 WRITE PRINT-LINE FROM HEAD-LINE
000474 015 AFTER ADVANCING TOP-OF-PAGE LINES.
000475 015 ADD 1 TO UNCOUNT.
000476 015 IF PT EQUALS 1
000477 015 MOVE '1' TO BE MOVED.
000478 015 MOVE '1' TO LINE-1 IN LINE1
000479 015 MOVE '2' TO LINE-2 IN LINE2
000480 015 ELSE
000481 015 MOVE '1' PREPO ' TO NAME IN LINE2
000482 015 MOVE ZERO TO STR IN LINE1
000483 015 MOVE ZERO TO PAX IN LINE2
000484 015 MOVE '3' TO LINE-1 IN LINE1
000485 015 MOVE '4' TO LINE-2 IN LINE2.
000486 015 MOVE STONA(PT, 1) TO STON-A0 IN LINE1.
000487 015 MOVE STONA(PT, 2) TO STON-B0 IN LINE1.
000488 015 MOVE STONA(PT, 3) TO STON-B1 IN LINE1.
000489 015 MOVE STONA(PT, 4) TO STON-A1 IN LINE1.
000490 015 MOVE STONA(PT, 5) TO STON-J1 IN LINE1.
000491 015 MOVE STONA(PT, 6) TO STON-B2C IN LINE1.
000492 015 MOVE STONA(PT, 7) TO STON-B2D IN LINE1.
000493 015 MOVE STONA(PT, 8) TO STON-A2C IN LINE1.
000494 015 MOVE STONA(PT, 9) TO STON-A2D IN LINE1.
000495 015 MOVE STONA(PT, 10) TO STON-J2D IN LINE1.
000496 015 MOVE STONA(PT, 11) TO STON-J2C IN LINE1.
000497 015 MOVE STONA(PT, 12) TO STON-A3C IN LINE1.
000498 015 MOVE STONA(PT, 13) TO STON-J3C IN LINE1.
000499 015 MOVE STONA(PT, 14) TO STON-J7C IN LINE1.
000500 015 MOVE STONA(PT, 15) TO STON-M7C IN LINE1.
000501 015 MOVE STONA(PT, 16) TO STON-UE IN LINE1.
000502 015 MOVE STONA(PT, 17) TO STON-AS IN LINE1.
000503 015 IF PT EQUALS 1
000504 015 WRITE PRINT-LINE FROM LINE1 AFTER ADVANCING 2 LINES
000505 015 ELSE
000506 015 WRITE PRINT-LINE FROM LINE1 AFTER ADVANCING 1 LINES.
000507 015 MOVE SFTTA(PT, 1) TO SFTT-A0 IN LINE2.
000508 015 MOVE SFTTA(PT, 2) TO SFTT-B0 IN LINE2.
000509 015 MOVE SFTTA(PT, 3) TO SFTT-B1 IN LINE2.
000510 015 MOVE SFTTA(PT, 4) TO SFTT-A1 IN LINE2.
000511 015 MOVE SFTTA(PT, 6) TO SFTT-B2C IN LINE2.

```

••UNCLASSIFIED••

MRG PRE-PROCESSOR PROGRAM POMCUSHOLL

••UNCLASSIFIED••

```

000512 015 MOVE SQFTA(PT, 7) TO SQFT-B2D IN LINE2.
000513 015 MOVE SQFTA(PT, 8) TO SQFT-A2C IN LINE2.
000514 015 MOVE SQFTA(PT, 9) TO SQFT-A2D IN LINE2.
000515 015 MOVE SQFTA(PT, 12) TO SQFT-A3C IN LINE2.
000516 015 MOVE SQFTA(PT, 16) TO SQFT-NSUA.
000517 015 MOVE SQFTA(PT, 17) TO SQFT-VEN IN LINE2.
000518 015 WRITE PRINT-LINE FROM LINE2.
000519 015 ENTER FORTRAN SDFTU SUBROUTINE.
000519 015 F1SAVE.
000520 015
000521 015 MOVE SRC19 IN LOG-REC TO SRC1-9 IN LINE1.
000522 015 MOVE SRC11-12 TO SRCPAR IN LINE1.
000523 015 MOVE LEVCO TO ULC IN LINE1.
000524 015 MOVE UTC IN LOG-REC TO UTC IN LINE1.
000525 015 MOVE UNSTR TO STR IN LINE1.
000526 015 MOVE UIC IN LOG-REC TO UIC IN LINE2.
000527 015 MOVE PERTO IN LOG-REC TO PAX IN LINE2.
000528 015 MOVE 2 TO PT.
000529 015 PERFORM ZERO-ARRAYS VARYING CAT FROM 1 BY 1
000530 015 UNTIL CAT IS GREATER THAN 18.
000531 015 MOVE 1 TO PT.
000532 015 PERFORM ZERO-ARRAYS VARYING CAT FROM 1 BY 1
000533 015 UNTIL CAT IS GREATER THAN 18.
000534 015 GO TO LOAD-F5.
000535 015 ZERO-ARRAYS.
000536 015 MOVE ZEROS TO STONAI(PT, CAT) SQFTA(PT, CAT).
000537 015 LAST-SUM.
000538 015 PERFORM PRINT-SUM.
000539 015 ENTER FORTRAN EOF12 SUBROUTINE.
000540 015 END-LOAD.
000541 015 EXIT.
000542 015 KEY-ERR.
000543 015 MOVE F5KEY TO NUMB.
000544 015 MOVE 'INVALID KEY FOR FIF2 FILE' TO MESSAGE.
000545 015 MOVE ERROR-WORK TO PRINT-LINE.
000546 015 PRINT-OUT.
000547 015 WRITE PRINT-LINE.
000548 015 CLOSE-FILES.
000549 015 IF LOAD5 CLOSE F5 USING F5-INFO.
000550 015 MOVE 'F5 INFORMATION AREA' TO PRINT-LINE.
000551 015 PERFORM PRINT-OUT.
000552 015 MOVE CORRESPONDING F5-INFO TO F5-INFORMATION.
000553 015 MOVE F5-INFORMATION TO PRINT-LINE.
000554 015 PERFORM PRINT-OUT.
000555 015 CLOSE PRINT-FILE.
000556 015 STOP RUN.

```

END ELT.

MRG MRG PROGRAM GEO ••UNCLASSIFIED•• L.O

```

BELT,L 58CAMP,GEO
ELT007 RL1870 08/16-07:43:31-120,)
000001 016 PARAMETER NG=9000
000004 016 IMPLICIT INTEGER (A-Z)
000003 016 REAL ACC
000004 016 DIMENSION THIR(7),AVAIL(7),DEST(7,2),AVT(7),
000005 016 * XTPSN(20),XDSTG(20),XDSTN(20),XAVL(20),XRDD(20),XMODE(20),
000006 016 * XTHR(20),XORG(20),XORGN(20)
000007 016 DIMENSION SRC(2),UNTD(2),MDF1(17)
000008 016 EQUIVALENCE (MDF1(1),SRC), * 2A6
000009 016 (MDF1(3),UIC), * A6
000010 016 (MDF1(4),TPSN), * 15
000011 016 (MDF1(5),ROB), * A1
000012 016 (MDF1(6),G024), * A6
000013 016 (MDF1(7),G024), * A6
000014 016 (MDF1(8),ONOD), * 13
000015 016 (MDF1(9),DNOD), * 13
000016 016 (MDF1(10),RDD), * 13
000017 016 (MDF1(11),AVL), * 13
000018 016 (MDF1(12),MODE), * A1
000019 016 (MDF1(13),THR), * 11
000020 016 (MDF1(14),MCOE), * 11
000021 016 (MDF1(15),UNTD), * 2A6
000022 016 (MDF1(17),FPSAV), * 12
000023 016 DIMENSION FASREC(52),F(52),LOCNAM(2),FNAME(3)
000024 016 * LIST(3,NG),FIRST(2),LAST(2)
000025 016 EQUIVALENCE (F,FASREC)
000026 016 EQUIVALENCE (LN1,LOCNAM(1)),(LN2,LOCNAM(2))
000027 016 C
000028 016 DATA FIRST/2*/
000029 016 DATA FNAME/58BIFAS
000030 016 *, 68B1GEO
000031 018 *, 68B1NODE/
000032 016 DATA IFL/17/
000033 016 DIMENSION FILE(2)
000034 016 DATA FILE /'GEO','NODE'/
000035 016 DATA T/***** B/***** E/***** R/*****
000036 016 * ONE/***** BLANK/ * MZERO/O/ M/*****
000037 016 * MED/***** DC/***** FOUR/***** XB/*****
000038 019 * TWO/***** THREE/***** NINE/*****
000039 016 C
000040 016 DIMENSION KEY(13)
000041 016 DATA KEY /1,1,36,0,0,1, * SRC(1)
000042 016 * 2,1,36,0,0,2, * SRC(2)
000043 016 * 99999/
000044 016 LOGICAL ERROR
000045 016 C
000046 016 C DEFINE CHARACTER FUNCTIONS
000047 016 C
000048 016 DEFINE C1(X) = FLD(0,6,X)
000049 016 DEFINE C2(X) = FLD(6,6,X)
000050 016 DEFINE C3(X) = FLD(12,6,X)
000051 016 DEFINE C4(X) = FLD(18,6,X)
000052 016 DEFINE C5(X) = FLD(24,6,X)
000053 016 DEFINE C6(X) = FLD(30,6,X)
000054 016 DEFINE C12(X) = FLD(0,12,X)
000055 016 DEFINE C13(X) = FLD(0,18,X)

```

AD-A048 554

ARMY CONCEPTS ANALYSIS AGENCY BETHESDA MD
COMPUTER ASSISTED MATCH PROGRAM (CAMP), (U)
AUG 76 G L MARTIN, E R MONTAGNE
CAA-D-76-5

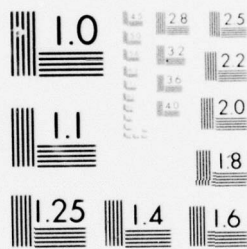
F/G 15/7

UNCLASSIFIED

NL

3 OF 4
AD-A048554





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

MRG PROGRAM GEO

••UNCLASSIFIED••

```

000056 016 DEFINE C14(X) = FLD(0,24,X)
000057 016 DEFINE C15(X) = FLD(0,30,X)
000058 016 DEFINE C23(X) = FLD(6,12,X)
000059 016 DEFINE C24(X) = FLD(6,18,X)
000060 016 DEFINE C25(X) = FLD(6,24,X)
000061 016 DEFINE C26(X) = FLD(6,30,X)
000062 016 DEFINE C34(X) = FLD(12,12,X)
000063 016 DEFINE C35(X) = FLD(12,18,X)
000064 016 DEFINE C36(X) = FLD(12,24,X)
000065 016 DEFINE C45(X) = FLD(18,12,X)
000066 016 DEFINE C46(X) = FLD(18,18,X)
000067 016 DEFINE C56(X) = FLD(24,12,X)
000068 016 C
000069 016 C
000070 016 C CREATES MOFI FILE IN SRC SEQUENCE
000071 016 C
000072 016 C ENTER NUMBER OF THEATERS
000073 016 READ(5,1000) NTH
000074 016 1000 FORMAT(11)
000075 016 C ENTER DESTINATION GEO AND NODE, AVAIL DAY
000076 016 READ(5,1001) (THTR(K),AVAIL(K),DEST(K,1),DEST(K,2),AVT(K),
000077 016 * K = 1,NTH)
000078 016 1001 FORMAT(11,1X,13,1X,4X,1X,13,1X,13)
000079 016 C ENTER TPSN LIST FOR EXCEPTIONS
000080 016 READ(5,1020) NTPSNX
000081 016 IF(NTPSNX.EQ.0) GO TO 1109
000082 016 READ(5,1002) (XTPSN(I),XDSGT(I),XAVL(I),XRDD(I),XMODE(I),
000083 016 * XTHR(I),XORGG(I),XORGN(I),I=1,NTPSNX)
000084 016 1002 FORMAT(15,1X,4X,3(1X,13),1X,4X,1X,13,1X,13)
000085 016 1020 FORMAT(12)
000086 016 C
000087 016 C PRINT OUT INPUT
000088 016 1109 PRINT 1110, NTH,NTPSNX
000089 016 1110 FORMAT(' NUMBER OF THEATERS ',12/
000090 016 * ' NUMBER OF EXCEPTIONS ',12)
000091 016 PRINT 1120, (THTR(K),AVAIL(K),DEST(K,1),DEST(K,2),AVT(K),K=1,NTH)
000092 016 1120 FORMAT(' THTR AVAIL DEST(G) DEST(N) AVL(T)'/
000093 016 * (3X,11,3X,13,3X,4X,4X,13,3X,13))
000094 016 IF(NTPSNX.EQ.0) GO TO 1131
000095 016 PRINT 1130, (XTPSN(I),XDSGT(I),XAVL(I),XRDD(I),XMODE(I),
000096 016 * XTHR(I),XORGG(I),XORGN(I),I=1,NTPSNX)
000097 016 1130 FORMAT(' TPSN DEST(G) DEST(N) AVL RUD MUDE THR',
000098 016 * ' ORIG(G) ORIG(N)'/
000099 016 * (1X,15,3X,4X,4X,13,5X,13,2X,13,4X,4X,1X,4X,4X,5X,13))
000100 016 C ERROR CHECK
000101 016 1131 IF(NTH.GT.0.AND.NTH.LE.7) GO TO 1150
000102 016 PRINT 1140, NTH
000103 016 1140 FORMAT(1X,12,' THEATERS INPUT. MUST BE BETWEEN 0 - 7')
000104 016 IER = 1

```

MRG PROGRAM GEO

MRG PROGRAM GEO

MRG PROGRAM GEO

```

000113 016 SU IG=IG+1
000114 016 READ(15,1,END=60) (LIST(1,IG),I=1,3)
000115 016 GOTOSO
000116 016 C
000117 016 C ENTER NODE TABLE (LIST)
000118 016 C LAST(1)=IG
000119 016 IG=IG-1
000120 016 FIRST(2)=IG
000121 016 IR=IG
000122 016 PRINT 95,IR,FNAME(2)
000123 016 95 FORMAT(18,' RECORDS ON FILE '2A6)
000124 016 C
000125 016 70 IG=IG+1
000126 016 READ(16,1003,END=79) LIST(1,IG),LIST(3,IG),LIST(2,IG)
000127 016 1003 FORMAT(A6,6X,A6,3X,13)
000128 016 C46(LIST(1,IG)) = 0
000129 016 1 FORMAT(12A6)
000130 016 GOTO70
000131 016 75 PRINT 76, IG
000132 016 76 FORMAT(16,' ENTRIES IN GEO AND NODE. CAN ONLY HAVE 9000')
000133 016 STOP
000134 016 79 CONTINUE
000135 016 IF(IG.GT. 9001) GO TO 75
000136 016 LAST(2)=IG
000137 016 IR=IG-IR-1
000138 016 PRINT 95,IR,FNAME(3)
000139 016 C
000140 016 C
000141 016 PRINT 97
000142 016 97 FORMAT(1,' NO FIND AVE ACC/U')
000143 016 C NO FIND ACTIVE NO FIND RESERVE'
000144 016 C
000145 016 C SET TO SORT OUTPUT ON SRC
000146 016 CALL SOPEN3(18100,5990,17,2,KEY)
000147 016 C MAIN LOOP
000148 016 C
000149 016 100 CONTINUE
000150 016 READ(1FL,END=900) FASREC
000151 017 C THEATER CODE
000152 016 THR = C1(F(8))-48
000153 016 DO 101 ITH = 1,NTH
000154 016 IF(THR.EQ. THTR(1TH)) GO TO 102
000155 016 101 CONTINUE
000156 016 GO TO 100
000157 016 C SKIP IF UNIT IS DISPLAY ONLY
000158 016 102 IF(C56(F(121)).NE. DC) GO TO 100
000159 016 C IF SRC IS BLANK /TDA UNIT/ DO NOT CONSIDER
000160 016 SRC(1) = C26(F(23))*2**6 + C1(F(24))
000161 016 IF(SRC(1).EQ. BLANK) GO TO 100
000162 016 SRC(2) = C24(F(24))*2**16 + C6(F(24))*2**12 + C1(F(25))*2**6
000163 016 C
000164 016 COMPO = C3(F(1))
000165 016 UIC = C46(F(1))*2**18 + C13(F(41))
000166 016 IF(COMPO.EQ. FOUR.AND. C12(UIC).EQ. X8) GO TO 100
000167 016 IF(COMPO.EQ. ONE.AND. C4(F(14)).EQ. B) GO TO 100
000168 016 KU = KU + 1
000169 016 C PROGRESS REPORT

```

MRG PROGRAM GEO

MRG PROGRAM GEO

MRG PROGRAM GEO

```

000170 016 IF(KU/500*500*EQ*U) PRINT 99,KU,KAK,KAN,KR,KRN,ACC
000171 016 DECODE(10,1004,F(30)) TPSN
000172 016 1004 FORMAT(5X,15)
000173 016 C
000174 016 C
000175 016 C
000176 016 C
000177 016 C COMPUTE /MCODE/
000178 016 C O=IN COUNTRY, 1=PREPO, 2=CONUS ORIENTED UN NATO, 3=CONUS OTHER
000179 016 C
000180 016 C ROB = BLANK
000181 016 C MCODE = 3
000182 016 IF(ICOMPO *NE* ONE) GO TO ZUD
000183 016 P3 = C6(F(22))
000184 016 IF(P3*EQ*E) C1(ROB) = P3
000185 016 IF(P3 *EQ* B) C1(ROB) = E
000186 016 P2 = C5(F(22))
000187 016 IF(P2 *EQ*E) C1(ROB) = P2
000188 016 IF(C46(F(22)) *EQ* MED) C1(ROB) = M
000189 016 IF(ROB *NE* BLANK) MCODE = 1
000190 019 C CHECK ADCOL FIELD - POS 119
000191 020 IF(C5(F(40)) *EQ* E *OR* C5(F(20)) *EQ* NINE) MCODE = 0
000192 016 IF(MCODE *NE* 1) ROB = BLANK
000193 016 C
000194 016 C ACTIVE UNIT
000195 016 C LOCCO STNM
000196 016 B4
000197 016 KAKA+1
000198 016 LN1 = C46(F(26))
000199 016 C13(LN1) = C46(F(15))
000200 016 IF(C1(LN1)*GE*48 *AND* C1(LN1)*LE*57) C1(LN1) = BLANK
000201 016 LN2 = F(27)
000202 016 LG = LOCATE(LOCNAM,1)
000203 016 IF(LG*LE*U) KAN=KAN+1
000204 016 C IF(LG *EQ* 0) PRINT 91, FILE(1),LOCNAM,UIC
000205 016 C LOCCO
000206 016 C13(LOCCO) = C13(LN1)
000207 016 LN = LOCATE(LOCCO,2)
000208 016 IF(LN*GT*0) GOT0500
000209 016 C
000210 016 C DIAGNOSTIC
000211 016 PRINT 91, FILE(2),LOCNAM,LOCCO,UIC
000212 016 91 FORMAT(' **ACTIVE UNIT AREA NOT FOUND IN ',A5,'FILE ',
000213 016 * 3X,Z46,Z(2X,A6))
000214 016 C
000215 016 C GOT0500
000216 016 C
000217 016 C
000218 016 C
000219 016 C
000220 016 C RESERVE UNIT
000221 016 ZUD CONTINUE
000222 016 KR=KR+1
000223 016 C
000224 016 C MBLOC MBSTA
000225 016 LN1 = C6(F(16))*2**30 + C12(F(17))*2**18 + C56(F(17))*2**6
000226 016 * C1(F(18))

```


MRG PROGRAM GEU **UNCLASSIFIED**

```

000227 016 IF(C1(LN1)*GE*48 .AND. C1(LN1)*LE*57) C1(LN1) = BLANK
000228 016 LN2 = C26(F181)*2**6 + C1(F191)
000229 016 LG = LOCATE(LOCNAM,1)
000230 016 C IF(LG.EQ.0) PRINT 92, FILE(1),LOCNAM,UIC
000231 016 IF(LG.LE.0) KRN=KRN+1
000232 016 C
000233 016 C MBLOC
000234 016 C13(MBLOC) = C13(LN1)
000235 016 LN = LOCATE(MBLOC,2)
000236 016 IF(LN.GT.0) GOT0500
000237 016 C
000238 016 C LOCCO
000239 016 C13(LOCCO) = C46(F151)
000240 016 IF(C1(LOCCO)*GE*48 .AND. C1(LOCCO)*LE*57) C1(LOCCO) = BLANK
000241 016 LN = LOCATE(LOCCO,2)
000242 016 IF(LN.GT.0) GOT0500
000243 016 C
000244 016 C DIAGNOSTIC
000245 016 PRINT 92, FILE(2),LOCNAM,LOCCO,UIC
000246 016 92 FORMAT(' *** RESERVE UNIT AREA NOT FOUND IN ',AS,'FILE ',
000247 016 * 3X,2A6,2(2X,A6))
000248 016 C
000249 016 C
000250 016 C
000251 016 C
000252 016 C
000253 016 C BUILD MOVEMENT DATA FILE RECORD
000254 016 500 CONTINUE
000255 016 C
000256 016 C
000257 016 C
000258 016 C RDD, AVAIL
000259 016 DECODE(6,1005,F(7)) RDD
000260 016 1005 FORMAT(2X,13)
000261 016 AVL = AVAIL(1TH)
000262 016 IF(TPSN.LT. 20000 .AND. (COMPO.EQ.TWO .OR. COMPO.EQ.THREE))
000263 016 * AVL = AVT(1TH)
000264 016 C
000265 016 C MODE, ULC
000266 016 MODE = 'P'
000267 016 IF(MCODE.EQ.0) MODE = 'Z'
000268 016 IF(MCODE.EQ.1) MODE = 'A'
000269 016 ULC = C6(F34)*2**30 + C12(F35)*2**24
000270 016 C
000271 016 C UNTDS (1-12)
000272 016 UNTDS(1) = C36(F35)*2**12 + C12(F36)
000273 016 UNTDS(2) = C36(F36)*2**12 + C12(F37)
000274 016 C
000275 016 C ORIGIN: G024, ARLOC (1-2) CK MATCH RESULTS
000276 016 IF(LN.EQ.0) GO TO 950
000277 016 IF(LG.EQ.0) LG=LN
000278 016 C
000279 016 G024 = LIST(3,LG)
000280 016 ONOD = LIST(2,LN)
000281 016 C
000282 016 C DESTINATION
000283 016 G024 = DEST(1TH,1)

```

MRG PROGRAM GEU **UNCLASSIFIED**


```

000284 016 DNOD = DEST(ITH,2)
000285 016 C
000286 016 IF(MCODE .NE. 0) GO TO 1040
000287 016 C IN COUNTRY = SET ORIGIN TO DESTINATION. NO EXCEPTION LIST
000288 016 GO24 = GO24
000289 016 GO TO 1090
000290 016 C SEE IF UNIT IS ON EXCEPTION LIST
000291 016 1040 DO 1050 I = 1,NTPSN
000292 016 IF(TPSN .EQ. XTPSN(I)) GO TO 1060
000293 016 1050 CONTINUE
000294 016 GO TO 1090
000295 016 1060 IF(XDSTG(I) .NE. BLANK) GO24 = XDSTG(I)
000296 016 IF(XDSTN(I) .NE. MZERO) DNOD = XDSTN(I)
000297 016 IF(XRDO (I) .NE. MZERO) RDO = XRDO (I)
000298 016 IF(XAVL (I) .NE. MZERO) AVL = XAVL (I)
000299 016 IF(XMODE(I) .NE. BLANK) MODE = XMODE(I)
000300 016 IF(XTHR (I) .NE. MZERO) THR = XTHR(I)
000301 016 IF(XORGG(I) .NE. BLANK) GO24 = XORGG(I)
000302 016 IF(XORGN(I) .NE. MZERO) DNOD = XORGN(I)
000303 016 C WRITE RECORD
000304 016 C DEBUG TO UNIT 7
000305 016 1090 WRITE(7,1100) MDF1
000306 016 1100 FORMAT(1X,2A6,1X,A6,1X,15,1X,A1,2(1X,A6),4(1X,13),1X,A1,2(1X,11),
000307 016 1X,2A6,1X,A1)
000308 016 CALL SRREL(MDF1,17)
000309 016 ACC = ((KU-1)*ACC+KC)/KU
000310 016 KC=0
000311 016 GOT0100
000312 016 C
000313 016 C
000314 016 900 ENDFILE 21
000315 016 PRINT 99,KU,KA,KAN,KR,KRN,ACC,KF
000316 016 99 FORMAT(I)
000317 016 C SORT MDF1 ACCORDING TO SRC
000318 016 980 CALL SSORT
000319 016 990 CALL SRRET(MDF1,17,$995)
000320 016 WRITE(29) MDF1
000321 016 GO TO 990
000322 016 995 ENDFILE 29
000323 016 IF(ERROR) RETURN 0
000324 016 STOP
000325 016 C
000326 016 C
000327 016 C CAN'T OUTPUT THIS MDF = SAVE FAS RECORD FOR REPROCESSING
000328 016 950 KU = KU - 1
000329 016 KF = KF + 1
000330 016 ERROR = .TRUE.
000331 016 GO TO 100
000332 016 C
000333 016 C
000334 016 C .....
000335 016 FUNCTION LOCATE(ARR,N)
000336 016 DIMENSION ARR(2)
000337 016 C
000338 016 LOCATE=0
000339 016 HI=LAST(N)
000340 016 LO=FIRST(N)

```

```
000341 016 C
000342 016 C BINARY SEARCH
000343 016 C
000344 016 100 LP=LO*(HI-LO)/2
000345 016 99 FORMAT()
000346 016 KC=KC+1
000347 016 1 FORMAT(2X,2A6,2X,2A6,16)
000348 016 IF(ARR(1)-LIST(1,LP)) 200,110,400
000349 016 110 IF(M*EG*2) GOTO300
000350 016 IF(ARR(2)-LIST(2,LP)) 200,300,400
000351 016 C
000352 016 C HIGH
000353 016 200 IF(LP=LO*LE.1) RETURN
000354 016 HI=LP
000355 016 GOTO100
000356 016 C
000357 016 C HIT
000358 016 300 LOCATE=LP
000359 016 RETURN
000360 016 C
000361 016 C LOW
000362 016 400 IF(HI=LP*LE.1) RETURN
000363 016 LO=LP
000364 016 GOTO100
000365 016 END
```

END ELT.

••UNCLASSIFIED•• L.O

MRG PROGRAM LOG

BMGG

08LT, L 58CAMP, LOG

ELT007 RL1870 08/16-07:43:36-10,1

000001 000 IMPLICIT INTEGER (A-X)
000002 000 DIMENSION SRC(2), UNTDS(2), MDF1(17), MDF1A(20)
000003 000 EQUIVALENCE (MDF1, MDF1A)
000004 000 EQUIVALENCE (MDF1(1), SRC), @ 2A6

000005 000 (MDF1(3), UIC), @ A6
000006 000 (MDF1(4), TPN), @ 15
000007 000 (MDF1(5), RUB), @ A1
000008 000 (MDF1(6), G024), @ A6
000009 000 (MDF1(7), G024), @ A6
000010 000 (MDF1(8), G024), @ 13

000011 000 (MDF1(9), DNOU), @ 13
000012 000 (MDF1(10), RUU), @ 13
000013 000 (MDF1(11), AVL), @ 13
000014 000 (MDF1(12), MOUE), @ A1
000015 000 (MDF1(13), THR), @ 11
000016 000 (MDF1(14), MCODE), @ 11

000017 000 (MDF1(15), UNTDS), @ 2A6
000018 000 (MDF1(17), FPSAV), @ 12

000019 000 DIMENSION P(22,4)
000020 000 EQUIVALENCE (P, TUCHA), (P(1,3), POMCUS)
000021 000 DIMENSION TUCHA(22,2), TSRC(2), TNAM(2)

000022 000 EQUIVALENCE (TUCHA(1,1), TSRC), (TUCHA(1,2), PUIC),
000023 000 (TUCHA(3,1), ULC), (TUCHA(3,2), TNAM),
000024 000 (TUCHA(4,1), UTC),

000025 000 (TUCHA(5,2), PAX),
000026 000 (TUCHA(6,2), VEHNAT),
000027 000 (TUCHA(7,2), NSDNAT),
000028 000 (TUCHA(8,2), NSDOU),
000029 000 (TUCHA(9,2), VEHOU),
000030 000 (TUCHA(10,2), ZNVOUT),
000031 000 (TUCHA(11,2), NSDOVC),
000032 000 (TUCHA(12,2), NSDOVN),
000033 000 (TUCHA(13,2), VEHVOC),
000034 000 (TUCHA(14,2), ZNVHVN),
000035 000 (TUCHA(15,2), ZNVQVN),
000036 000 (TUCHA(16,2), ZNVVOC),
000037 000 (TUCHA(17,2), VEHBC),
000038 000 (TUCHA(18,2), ZNVBC)

000039 000 DIMENSION POMCUS(22,2)
000040 000 EQUIVALENCE

000041 000 (POMCUS(6,1), YVEHNA), (POMCUS(5,2), PAXP),
000042 000 (POMCUS(7,1), YNSDNA), (POMCUS(6,2), VEHNAP),
000043 000 (POMCUS(8,1), YNSDOU), (POMCUS(7,2), NSDNAP),
000044 000 (POMCUS(9,1), YVEHOU), (POMCUS(8,2), NSDOUP),
000045 000 (POMCUS(10,1), YNVOUT), (POMCUS(9,2), VEHQUP),
000046 000 (POMCUS(11,1), YNSVOC), (POMCUS(11,2), NSOVCP),
000047 000 (POMCUS(12,1), YNSOVN), (POMCUS(12,2), NSOVNP),
000048 000 (POMCUS(13,1), YVHVCV), (POMCUS(13,2), VHOVCP),
000049 000 (POMCUS(14,1), YVHVCV), (POMCUS(14,2), VHOVNP),
000050 000 (POMCUS(15,1), YNVQVN),
000051 000 (POMCUS(16,1), YNVVOC), (POMCUS(17,2), VEHBCP),
000052 000 (POMCUS(17,1), YVEHBC),
000053 000 (POMCUS(18,1), YNVBC)

000054 000 DIMENSION PORTS(2), PLANID(7), MORSW(6), MORSWP(6)
000055 000 EQUIVALENCE (MORSW(1), ASUPLY), (MORSWP(1), PASUPL),


```
000056 000 * (MORSW(2), BULK), (MORSW(2), PBULK),
000057 000 * (MORSW(3), OVER), (MORSW(3), POVER),
000058 000 * (MORSW(4), OUT), (MORSW(4), POUT),
000059 000 * (MORSW(5), NAT), (MORSW(5), PNAT),
000060 000 * (MORSW(6), ACAMMO), (MORSW(6), PACAMM),
000061 000 DIMENSION SEQ(7), THTR(7), TSAR(2), ZASR(2), THTRID(7),
000062 000 * ICT(7)
000063 000 LOGICAL ENDT, ENDP, ENDM, ERROR, FIND
000064 000 DIMENSION SSRC(4,99)
000065 000 DIMENSION IZERO(44)
000066 000 DATA IZERO /44*0/
000067 000 DIMENSION KEY(7), KEY1(13)
000068 000 DATA KEY /3,1,36,0,0,1, * UIC
000069 000 * 99999/
000070 000 DATA KEY1 /1,1,36,0,0,1, * SRC(1)
000071 000 * 2,1,36,0,0,2, * SRC(2)
000072 000 * 99999/
000073 000 DATA MD/'Z', PAVL/999/, PRDD/0/
000074 000 C
000075 000 DEFINE C3(X) = FLD(12,6,X)
000076 000 C
000077 000 READ(5,1) IN,NTH,THTR,THTRID,PLANID
000078 000 * FORMAT(2(1,1X),2(7(1,1X),7A1) * DEBARKATON AND ASR(LBS)
000079 000 READ(5,2) PORTS,ZBASR,ZAASK
000080 000 * FORMAT(2(4,1X),4(F6,0,1A1)
000081 000 READ(5,3) NSRCX
000082 000 * FORMAT(12)
000083 000 IF(NSRCX *EQ* 0) GO TO 44
000084 000 READ(5,4) ((SSRC(I,J),I=1,4),J=1,NSRCX)
000085 000 * FORMAT(46)
000086 000 C SORT SRC INPUT - MATCH UPMDFI INPUT
000087 000 CALL SOPEN(1,40,342,4,2,KEY1)
000088 000 DO 4) I = 1,NSRCX
000089 000 * CALL SRREL(SSRC(1,1),4)
000090 000 CALL SSORT
000091 000 I = 1
000092 000 * CALL SRREL(SSRC(1,1),4,344)
000093 000 I = I + 1
000094 000 GO TO 43
000095 000 CONTINUE
000096 000 C
000097 000 IF(IN *LE* 0 *OR* IN *GT* 7) IER = 1
000098 000 IF(NTH *LE* 0 *OR* NTH *GT* 7) IER = IER + 2
000099 000 DO 5 I = 1,IN
000100 000 * IF(THTRID(I) *EQ* 0) IER = IER + 4
000101 000 IF(1 *GT* NTH) GO TO 5
000102 000 * IF(PLANID(I) *EQ* ' ') IER = IER + 8
000103 000 CONTINUE
000104 000 PRINT 6, IN,NTH
000105 000 * FORMAT(1, NUMBER OF THEATERS ',12/
000106 000 * ', NUMBER OF SETS ',12)
000107 000 PRINT 7
000108 000 * FORMAT(1, THTR SET PLANID')
000109 000 DO 8 I = 1,IN
000110 000 * J = THTRID(I)
000111 000 PRINT 9, THTR(1),THTRID(1),PLANID(J)
000112 000 * FORMAT(3X,11,4X,11,6X,A1)
```



```

000113 PRINT 10
000114 FORMAT(6X,'AIRPORT',6X,'SEAPORT',10X,'NON-POMCUS ASR',15X,
000115 * 'POMCUS ASR',/33X,'SUPPLY',9X,'AMMO',7X,'SUPPLY',9X,'AMMO')
000116 PRINT 11, PORTS,(ZBASR(1),ZAASR(1),I=1,2)
000117 FORMAT(8X,A4,YX,A4,5X,2(FY,3,5X,FY,3,3X))
000118 IF(NSRCX.EQ.0) GO TO 16
000119 PRINT 17, ((SSRC(I),I=1,4),J=1,NSRCX)
000120 FORMAT(/, SRC LIST,/12X,2A6,2X,2A6))
000121 IF(1ER.EQ.0) GO TO 98
000122 IF(MOD(1ER,2).NE.0) PRINT 12
000123 FORMAT(/, *****ERROR***** NUMBER OF THEATERS')
000124 1ER = 1ER/2
000125 IF(MOD(1ER,2).NE.0) PRINT 13
000126 FORMAT(/, *****ERROR***** NUMBER OF SETS')
000127 1ER = 1ER/2
000128 IF(MOD(1ER,2).NE.0) PRINT 14
000129 FORMAT(/, *****ERROR***** SET NUMBERS')
000130 1ER = 1ER/2
000131 IF(MOD(1ER,2).NE.0) PRINT 15
000132 FORMAT(/, *****ERROR***** PLANID ERROR')
000133 RETURN 0
000134 CONTINUE
000135 C
000136 C
000137 C READ MDF1, FIND MATCH ON TUCHA (AND POMCUS IF MCODE = 1). FILES IN SRC SEW
000138 C PRODUCE UNIT MORSA CARDS AND MDF2
000139 C 29 = MDF1, 28 = TUCHA, 27 = POMCUS, 26 = MDF2, 10 = MORSA CARDS
000140 C
000141 CALL SOPEN2(899,$430,20,3,KEY)
000142 POM = 1
000143 FIND = .TRUE.
000144 DO 999 I = 1,2
000145 C CONVERT TO TONS
000146 ZBASR(1) = ZBASR(1)/2000.
000147 ZAASR(1) = ZAASR(1)/2000.
000148 C
000149 READ(28,151) TUCHA
000150 READ(29,END=400) MDF1
000151 IF(ENDT) GO TO 180
000152 NU = NU + 1
000153 IF(SRC(1) = TSRC(1)) 180,120,150
000154 C CHECK ONLY 7,8,9 OF SRC(2)
000155 IF(FLD(0,18,5SRC(2)) = FLD(0,18,TSRC(2))) 180,200,150
000156 READ(28,151,END=1030) TUCHA
000157 FORMAT(A6,A5,3X,A3,2X,A5,18F6.1/A6,A5,1X,2A6,18I6)
000158 GO TO 110
000159 C NO MATCH FOR MDF1 UNIT ON TUCHA
000160 IF(SRC(1).NE.TSAV(1)) GO TO 185
000161 IF(SRC(2).NE.TSAV(2)) GO TO 185
000162 GO TO 190
000163 PRINT 181, SRC,UIC,TSRC,NU
000164 FORMAT(/, NO MATCH FOR SRC ',2A6','UIC ',A6,' ON TUCHA FILE ',2A6,
000165 * 16)
000166 TSAV(1) = SRC(1)
000167 TSAV(2) = SRC(2)
000168 ERROR = .TRUE.
000169 NCT = NCT + 1
000170 GO TO 100

```

UNCLASSIFIED

```
000170 000 C
000171 000 C   MATCHED SRC'S. IS THIS A PORCUS UNIT?
000172 000 200 IF(MCODE *EQ. 1) GO TO 300
000173 000 C
000174 000 C   GET THEATER POSITION
000175 000 215 DO 216 ITH = 1,IN
000176 000 IF(THR *EQ. THTR(ITH)) GO TO 217
000177 000 CONTINUE
000178 000 PRINT 2000, THTR,SRC,UIC,TPSN,THR
000179 000 2000 FORMAT(716/, NO THEATER MATCH FOR '2A6,1X,A6,1X,15,1X,13)
000180 000 IF(ENDM) GO TO 430
000181 000 GO TO 100
000182 000 C
000183 000 C   FIX UP SRC IF IT WAS CHANGED IN UPMDFI
000184 000 217 IF(FPSAV *EQ. 0) GO TO 218
000185 000 IF(SRC(1) *NE. SSRC(3,FPSAV)) GO TO 1050
000186 000 IF(SRC(2) *NE. SSRC(4,FPSAV)) GO TO 1050
000187 000 SRC(1) = SSRC(1,FPSAV)
000188 000 SRC(2) = SSRC(2,FPSAV)
000189 000 218 RD = 6
000190 000 ITH = THTRID(ITH)
000191 000 ICT(ITH) = ICT(ITH) + 1
000192 000 C IF TPSN < 20000 USE SECOND CHARACTER
000193 000 IF(TPSN *LT. 20000) RD = MOD(TPSN/1000,10)
000194 000 IF(TPSN *GE. 30000) RD = 7
000195 000 TAVL = AVL
000196 000 TRDD = RDD
000197 000 C SET FOR IN-COUNTRY
000198 000 IF(MCODE *EQ. 0) TAVL = 999
000199 000 IF(MCODE *EQ. 0) TRDD = 0
000200 000 C
000201 000 IF(.NOT. FIND) GO TO 209
000202 000 C SET W CARD DATA
000203 000 TPAX = PAX
000204 000 BULK = ZVEHBC + ZNVBC + .5
000205 000 OVER = ZVHOVC + ZVHOVN + ZNVOVN + ZNV0VC + ZNSOVC + ZNSOVN + .5
000206 000 OUT = ZVEHOU + ZNVOUT + ZNSDOU + .5
000207 000 NAT = ZVEHNA + ZNSDNA + .5
000208 000 FRN = THR*1000 + UNOD
000209 000 NSDAST = ZNSDNA + ZNSDOU + ZNSOVC + ZNSOVN + .5
000210 000 NSDASQ = NSDNAT + NSDOU + ZNSOVC + ZNSOVN
000211 000 ASUPLY = FLOAT(TPAX) * ZBASR(POM) + .5
000212 000 ACAMMO = FLOAT(TPAX) * ZAASR(POM) + .5
000213 000 C
000214 000 C WRITE A CARD
000215 000 SEQ(ITH) = SEQ(ITH) + 1
000216 000 WRITE(10,210) SEQ(ITH),PLANID(ITH),RD,UNTOS,ULC,MODE,TAVL,TRDD,
000217 000 * 6024,6024,6024,TPAX,TPSN
000218 000 210 FORMAT(J4,A,A1,A U',11,2A6,A3,A1,2X,2J3,6X,3A4,2J6,13X)
000219 000 C WRITE W CARD
000220 000 WRITE(10,220) SEQ(ITH),PLANID(ITH),MORSW,SRC,FRN,UIC,UIC
000221 000 FORMAT(J4,A,A1,A1,M,6J6,A6,A5,J4,A6,11X)
000222 000 IF(NSDAST + NSDASQ *EQ. 0) GO TO 250
000223 000 C WRITE F CARD
000224 000 WRITE(10,230) SEQ(ITH),PLANID(ITH),NSDAST,NSDASQ
000225 000 230 FORMAT(J4,A,A1,A1,FN',12X,2J6,48X)
000226 000 C
```

UNCLASSIFIED

```

000227 000 C WRITE OUT DATA FOR LAYIN
000228 000 250 TOTAL = 0
000229 000 DO 240 I = 1,6
000230 000 240 TOTAL = TOTAL + MURSW(I)
000231 000 WRITE(11,241) TAVL,UIC,MOD,SEQ(1TH),G024,G024,MODE,TRDD,DNOD,UNOD,
000232 000 * BULK,OVER,OUT,NAT,NSDASQ,TOTAL,ASUPPLY,IZERO(1),TPAX,ACAMMO
000233 000 241 FORMAT(3X,J3,46,5X,11,5X,J6,244,A1,4X,1X,J3,2X,5X,10J5)
000234 000 C
000235 000 C PROCESSING POMCUS?
000236 000 IF(ENDM) GO TO 475
000237 000 C WRITE MDF2
000238 000 WRITE(26) MDF1,TUCHA,ASUPPLY,ACAMMO
000239 000 GO TO 100
000240 000 C
000241 000 C SAVE POMCUS UNITS
000242 000 300 MDFIA(18) = ULC
000243 000 MDFIA(19) = UTC
000244 000 MDFIA(20) = PAX
000245 000 NP = NP + 1
000246 000 CALL SRREL(MDFIA,20)
000247 000 GO TO 100
000248 000 C
000249 000 C END OF MDF1
000250 000 400 ENDM = .TRUE.
000251 000 PRINT 1011, ICT,SEQ,NCT
000252 000 POM = 2
000253 000 READ(27,151) P
000254 000 C SORT POMCUS UNITS ON UIC
000255 000 CALL SSORT
000256 000 430 CALL SRREL(MDFIA,20,$1000)
000257 000 IF(ENDP) GO TO 460
000258 000 FIND = .TRUE.
000259 000 C FIND MATCH ON POMCUS FILE
000260 000 440 IF(UIC = PUIC) 460,470,450
000261 000 450 READ(27,151,END=1040) P
000262 000 GO TO 440
000263 000 C NO MATCH
000264 000 460 PRINT 461, SRC,UIC,PUIC
000265 000 461 FORMAT(' NO MATCH FOR SRC ',2A6,' UIC ',A6,' ON POMCUS FILE ',A6)
000266 000 NCT = NCT + 1
000267 000 C WRITE ZERO RECORD, USE PAX,UTC,ULC FROM TUCHA
000268 000 FIND = .FALSE.
000269 000 TPAX = MDFIA(20)
000270 000 DO 462 I = 1,6
000271 000 MURSW(I) = 0
000272 000 462 MURSWP(I) = 0
000273 000 NSDAST = 0
000274 000 NSDASQ = 0
000275 000 PMSDST = 0
000276 000 PMSDSQ = 0
000277 000 IZERO(3) = ULC
000278 000 IZERO(4) = UTC
000279 000 IZERO(27) = TPAX
000280 000 GO TO 470
000281 000 C
000282 000 C PROCESS POMCUS UNIT - CREATE 2 SETS OF MORSA CARDS. SMOBSMOD
000283 000 C DATA ONLY FOR STUFF TO BE MOVED

```



```

000284 000 C
000285 000 470 ULC = MDF1A(18)
000286 000 UTC = MDF1A(19)
000287 000 C WRITE CARDS FOR STUFF TO BE MOVED
000288 000 GO TO 215
000289 000 C
000290 000 475 WRITE(10,476) SEQ(1:ITH),PLANID(1:ITH),ROB
000291 000 C
000292 000 IF(.NOT. FIND) GO TO 477
000293 000 C COMPUTE AMOUNT ALREADY THERE
000294 000 PBULK = YVEHBC + YNVBC + .5
000295 000 POVER = YVHOVC + YVHOVN + YNVGVN + YNVUVC + YNSOVC + YNSOVN + .5
000296 000 POUT = YVEHOU + YVOUT + YNSDOU + .5
000297 000 PNAT = YVEHNA + YNSDNA + .5
000298 000 PASUP = FLOAT(PAXP) * ZBASR(POM) + .5
000299 000 PACAMM = FLOAT(PAXP) * ZASR(POM) + .5
000300 000 PMSDT = YNSDNA + YNSDOU + YNSOVC + YNSOVN + .5
000301 000 PMSDSQ = NSDNAP + NSDOUP + NSOVC + NSOVNP
000302 000 C WRITE CARDS FOR STUFF ALREADY THERE
000303 000 477 SEQ(1:ITH) = SEQ(1:ITH) + 1
000304 000 WRITE(10,210) SEQ(1:ITH),PLANID(1:ITH),RD,UNTDs,ULC,MD,PAVL,PKDD,G024,
000305 000 * G024,G024,PAXP,TPSN
000306 000 WRITE(10,220) SEQ(1:ITH),PLANID(1:ITH),MORSWP,SRC,FRN,UTC,UIC
000307 000 IF(PMSDT + PMSDSQ .NE. 0) WRITE(10,230) SEQ(1:ITH),PLANID(1:ITH),
000308 000 * PMSDT,PMSDSQ
000309 000 WRITE(10,476) SEQ(1:ITH),PLANID(1:ITH),ROB
000310 000 476 FORMAT(J4,'A',A1,'FA',IX,A1,70X)
000311 000 C
000312 000 IF(.NOT. FIND) GO TO 480
000313 000 C MDF2 FILE SHOULD CONTAIN AMOUNTS TO BE MOVED
000314 000 C
000315 000 WRITE(26) MDF1,TUCHA,ASUPPLY,ACAMMO
000316 000 GO TO 430
000317 000 C
000318 000 C WRITE ZERO RECORD
000319 000 480 WRITE(26) MDF1,IZERO,ASUPPLY,ACAMMO
000320 000 GO TO 430
000321 000 C END OF MDF1 AND POMCUS UNITS
000322 000 C WRITE OUT SEQ TO BE USED IN NUR FOR REST OF MORSA CARDS
000323 000 1000 DO 1010 I = 1,7
000324 000 1010 MDF1(I) = SEQ(I)
000325 000 C SET MOD 50 IT WILL SORT OUT FIRST IN NUR
000326 000 MDF1(10) = -99999
000327 000 WRITE(26) MDF1,TUCHA,ASUPPLY,ACAMMO
000328 000 ENDFILE 26
000329 000 ENDFILE 10
000330 000 ENDFILE 11
000331 000 PRINT 1011, ICT,SEQ,NCT
000332 000 1011 FORMAT(' UNITS IN THEATER SETS ',717/
000333 000 * ' FINAL SEQUENCE NUMBERS ',717/
000334 000 * ' NO MATCHES ',17)
000335 000 IF(ERROR) RETURN 0
000336 000 STOP
000337 000 C
000338 000 C END OF TUCHA FILE. STILL HAVE MDF1 RECORDS LEFT
000339 000 1030 PRINT 1031
000340 000 1031 FORMAT(' END OF TUCHA FILE ')

```



```

000341      000      ENDT = .TRUE.
000342      000      GO TO 180
000343      000
000344      000      C
000345      000      C  END OF POMCUS FILE. STILL HAVE MDFIA RECORDS LEFT
000346      000      1040 PRINT 1041
000347      000      1041 FORMAT(' END OF POMCUS FILE ')
000348      000      ENDP = .TRUE.
000349      000      GO TO 460
000350      000      C
000351      000      C  BAD MATCH WITH SRC UPDATES
000352      000      1050 PRINT 1051, (MDF1(1),I=1,4),FPSAV,(SSRC(1),FPSAV),I=1,4)
000353      000      1051 FORMAT(' UNIT ',2A6,1X,A6,1X,15,' NO MATCH TO REVERT BACK TO '
000354      000      * 'ORIGINAL SRC. ENTRY NUMBER IS ',12/
000355      000      * 'OLD SRC IS ',2A6,' NEW SRC IS ',2A6)
000356      000      RETURN 0
000357      000      END

```

END ELT.

MRG PROGRAM UPMDFI **UNCLASSIFIED** L.O

```

BELT,L 58CAMP,UPMDFI
ELT007 RL1870 08/16-07:43:39-(10.)
000001 000 C THIS ROUTINE UPDATES SRC'S BASED ON AN INPUT TABLE - USED IF ERRORS
000002 000 C IN MATCHING UNITS IN LOG
000003 000 C IMPLICIT INTEGER(A=Z)
000004 000 DIMENSION MDI(17),SRC(4,100),KEY(13),KEY1(13)
000005 000 DATA KEY /1,1,36,0,0,1,
000006 000 2,1,36,0,0,2,
000007 000 99999/
000008 000 DATA KEY1/1,1,36,0,0,1,
000009 000 2,1,36,0,0,2,
000010 000 99999/
000011 000 LOGICAL END
000012 000 READ(5,1) NSRC W # OF UPDATE ENTRIES
000013 000 1 FORMAT(12)
000014 000 READ(5,2) ((SRC(J,1),J=1,4),I=1,NSRC) W OLD AND NEW
000015 000 2 FORMAT(46)
000016 000 END = .FALSE.
000017 000 C SRT INPUT SRC'S
000018 000 C MDI SHOULD BE IN SRC SORT (OUTPUT FROM GEU)
000019 000 CALL SOPEN(100,200,4,2,KEY)
000020 000 DO 110 I = 1,NSRC
000021 000 110 CALL SRREL(SRC(1,1),4)
000022 000 CALL SSORT
000023 000 I = 1
000024 000 210 CALL SRRET(SRC(1,1),4,$250)
000025 000 I = I + 1
000026 000 GO TO 210
000027 000 I = 1
000028 000 250 I = 1
000029 000 C FINAL MDI MUST BE IN SRC SORT
000030 000 CALL SOPEN(300,410,17,2,KEY1)
000031 000 READ(29,END=400) MDI
000032 000 IF(END) GO TO 330
000033 000 IF(MDI(1) = SRC(1,1)) 330,310,340
000034 000 IF(MDI(2) = SRC(2,1)) 330,320,340
000035 000 MDI(1) = SRC(3,1)
000036 000 MDI(2) = SRC(4,1)
000037 000 MDI(17) = I
000038 000 CALL SRREL(MDI,17)
000039 000 GO TO 300
000040 000 I = I + 1
000041 000 IF(1.GT. NSRC) GO TO 350
000042 000 GO TO 305
000043 000 END = .TRUE.
000044 000 GO TO 330
000045 000 CALL SSORT
000046 000 410 CALL SRRET(MDI,17,$500)
000047 000 WRITE(25) MDI
000048 000 GO TO 410
000049 000 ENDFILE 25
000050 000 STOP
000051 000 END
END ELT.
```

MRG PROGRAM UPMDFI

••UNCLASSIFIED••

DATE 081676

PAGE 114

MRG PROGRAM NUR

••UNCLASSIFIED•• L.O

END

MRG PROGRAM UPMDFI

••UNCLASSIFIED••

DATE 081676

PAGE 114

```

**UNCLASSIFIED**
SBCAMP.NUR
ELT007 RL1870 08/16-07:43:42-110.)
000001 006 IMPLICIT INTEGER (A=M)
000002 006 DIMENSION MDF2(63)
000003 006 DIMENSION SRC(2),MDF1(17)
000004 006 EQUIVALENCE (MDF1(1),SRC), @ 2A6
000005 006 (MDF1(3),UIC), @ 46
000006 006 (MDF1(4),TPSN), @ 15
000007 006 (MDF1(10),RDD), @ 13
000008 006 (MDF1(11),AVL), @ 13
000009 006 (MDF1(13),THR), @ 11
000010 006 (MDF1(14),MCOOE), @ 11
000011 006 DIMENSION TUCHA(22,2)
000012 006 EQUIVALENCE (MDF2,MDF1), (MDF2(18),TUCHA),
000013 006 EQUIVALENCE (MDF2(62),ASUPPLY), (MDF2(63),ACAMMO)
000014 006 DIMENSION CDAY(7),BLDUP(7),PRD(8,7),XLEVEL(3,7),LDAY(7),FDAY(7)
000015 006 XSAFE(3,7),GORG(20),NORG(20),XPCT(2,20),XCNSM(3,7,8)
000016 006 XASR(3,7,2),XRPLC(8,7),XSLOPE(3,7),PLANID(7),THTRID(7)
000017 006 XSTREN(7),SEQ(7),XRPLCE(7),XRS(3,7),DDAY(7),PER(7)
000018 006 RT(3),RD(4),RN(2,8),LVL(3),RDAY(7),ARLEV(3,7),AVDAY(7)
000019 006 DEST(2,7),KU(7),THTR(7),THEND(7),XTUN(210)
000020 006 EQUIVALENCE (XTON,XCNSM), (XTUN(169),XASR)
000021 006 DIMENSION SNOB(8),KEY(7)
000022 006 DATA RT/1HS,1HS,1HP/
000023 006 RD/1HG,1HA,1H2,1HF/
000024 006 RN/RESUPPLY DRY/
000025 006 RESUPPLY AMM/
000026 006 RESUPPLY POL/
000027 006 REPLACEMENTS/
000028 006 FILLER/
000029 006 PWRS DRY/
000030 006 PWRS AMMO/
000031 006 PWRS POL/
000032 006 DATA LVL/3HPKG,3HPKG,3HPOL/
000033 006 DATA PER/7,1,36,1,0,1,99999/ @ RDD SORT
000034 006 DIMENSION ADAY(7),XFR(7),BUAY(3,7),XBFR(3,7),MD(4),MOE(7)
000035 006 DATA MD/'P','S','A','2,/
000036 006 LOGICAL END
000037 006
000038 006 C COMPUTE NON-UNIT MOVEMENT REQUIREMENTS
000039 006 C
000040 006 C
000041 006 C
000042 006 C INITIALIZE
000043 006 C
000044 006 C THEATER DEFINITION
000045 006 READ(5,1001) IN,NTH,THTR,THTRID,PLANID
000046 006 FORMAT(2(1,1X),2(7,1,1X),7A1)
000047 006 C DAYS,PERIODS,LEVELS - BY SET
000048 006 READ(5,1002) (K,DDAY(K),RDAY(K),BLDUP(K),CDAY(K),ADAY(K),
000049 006 (PRD(L,K),L=1,8),(XLEVEL(L,K),L=1,3),(XSAFE(L,K),L=1,3),
000050 006 I = 1,NTH)
000051 006 1002 FORMAT(11,1X,13(13,1X),6(F3,0,1X))
000052 006 C BASE, FILLER, DEST - BY SET
000053 006 READ(5,1003) (K,(XRS(L,K),L=1,3),XFR(K),BDAY(L,K),L=1,3),
000054 006 (XBFR(L,K),L=1,3),DEST(L,K),L=1,2),AVDAY(K),I = 1,NTH)
000055 006 1003 FORMAT(11,1X,3(F9,0,1X),F4,0,1X,3(13,1X),3(F4,0,1X),A4,2(1X,13))

```



```

000056 C REPL (CAS) BY SET
000057 READ(5,1004) (K,(XRPCL(L,K),L=1,8),I=1,NTH)
000058 1004 FORMAT(11,1X,8(F5.0,1X))
000059 C ASR BY SET
000060 READ(5,1005) (K,(XASR(J,K,L),J=1,3),L=1,2),I=1,NTH)
000061 1005 FORMAT(11,1X,6(F6.0,1X))
000062 C CONSUMPTION - BY TYPE/SET
000063 READ(5,1006) (K,I,(XCNSM(J,K,L),L=1,8),J=1,3),I=1,NTH)
000064 1006 FORMAT(11,1X,8(F6.0,1X))
000065 C ORIGINS
000066 READ(5,1007) (GORG(L),NORG(L),(XPCT(J,L),J=1,2),L=1,20)
000067 1007 FORMAT(14,1X,13,1X,2(F4.0,1X))
000068 C
000069 PRINT 1100, IN,NTH
000070 1100 FORMAT(' NUMBER OF THEATERS ',I2/
000071 ' , NUMBER OF SETS ',I2)
000072 PRINT 1101, (THTR(I),I=1,IN)
000073 1101 FORMAT(' THTR SET',(3X,11,3X,11/))
000074 PRINT 1102
000075 1102 FORMAT(' SET PLAN UDAY RDAY BLOUP COAY ADAY 1 2 3 4 5,
000076 ' , 6 7 8 BLOUP LEVELS SAFE LEVELS')
000077 DO 1103 K = 1,NTH
000078 1103 PRINT 1104, K,PLANID(K),UDAY(K),RDAY(K),BLOUP(K),CDAY(K),
000079 ' , ADAY(K),(PRO(L,K),L=1,8),(XLEVEL(L,K),L=1,3),(XSAFE(L,K),
000080 ' , L=1,3)
000081 1104 FORMAT(2X,11,3X,A1,4X,13,2X,13,3X,13,2(2X,13),8(1X,13),
000082 ' , 2(1X,3(1X,F4.0)))
000083 PRINT 1105
000084 1105 FORMAT(' SET ',14X,'THEATER BASE',14X,'FILLER',5X,'FILLER',10X,
000085 ' , FILLER',25X,'DAY',9X,'DRY',9X,'AMMO',10X,'POL',7X,'PCT',7X,
000086 ' , 'DAYS',12X,'PCTS',9X,'DEST(G) DEST(N) AVL')
000087 DO 1106 K = 1,NTH
000088 1106 PRINT 1107, K,(XRS(L,K),L=1,3),XFR(K),(BDAY(L,K),L=1,3),
000089 ' , (XFR(L,K),L=1,3),(DEST(L,K),L=1,2),AVDAY(K)
000090 1107 FORMAT(2X,11,1X,3(1X,F12.3),2X,F5.3,2X,3(1X,1X),3(1X,F5.3),
000091 ' , 2X,4,2(5X,13))
000092 PRINT 1108
000093 1108 FORMAT(' REPLACEMENTS BY TIME PERIOD (MEN/1000/DAY)')
000094 DO 1109 K = 1,NTH
000095 1109 PRINT 1110, K,(XRPCL(L,K),L=1,8)
000096 1110 FORMAT(1X,11,1X,8(2X,F12.5))
000097 PRINT 1111
000098 1111 FORMAT(' ACCOMPANYING SUPPLY RATES (LBS/MAN)')
000099 ' , 5X, 'NON-PONCUS',33X,'POMCUS')
000100 DO 1112 K = 1,NTH
000101 1112 PRINT 1110, K,(XASR(J,K,L),J=1,3),L=1,2)
000102 PRINT 1113
000103 1113 FORMAT(' CONSUMPTION RATES (LBS/MAN/DAY) - BY TIME PERIOD')
000104 DO 1114 K = 1,NTH
000105 1114 PRINT 1115, K,(XCNSM(J,K,L),L=1,8),J=1,3)
000106 1115 FORMAT(2X,11,2X,'DRY',8(2X,F12.5),5X,'AMMO',8(2X,F12.5)/
000107 ' , 5X,'POL',8(2X,F12.5))
000108 PRINT 1116, (GORG(L),NORG(L),(XPCT(J,L),J=1,2),L=1,20)
000109 1116 FORMAT(' ORIG(N) DRY PCT AMMO PCT')
000110 ' (1X,4X,6X,13,6X,F5.3,5X,F5.3))
000111 DO 1117 I = 1,20
000112 XTOTI = XTOTI + XPCT(I,1)

```

```

000113 006 1117 XTOT2 = XTOT2 + XPCT(2,1)
000114 006 IER = 0
000115 006 TOT = XTOT1*100. + .5
000116 006 IF(TOT .EQ. 100) GO TO 1119
000117 006 IER = 1
000118 006 PRINT 1118, XTOT1
000119 006 1118 FORMAT(.....ERROR..... DRY PERCENTS ADD UP TO ',F5.3)
000120 006 1119 TOT = XTOT2*100. + .5
000121 006 IF(TOT .EQ. 100) GO TO 1121
000122 006 IER = 1
000123 006 PRINT 1120, XTOT2
000124 006 1120 FORMAT(.....ERROR..... AMMO PERCENTS ADD UP TO ',F5.3)
000125 007 1121 IF(IN .LE. 7) GO TO 1123
000126 007 PRINT 1122, IN
000127 007 IER = 1
000128 007 1123 IF(INTH .LE. 7) GO TO 1126
000129 007 PRINT 1127, NTH
000130 007 IER = 1
000131 007 1126 DO 1130 I = 1,IN
000132 007 1122 FORMAT(.....ERROR.....ILLEGAL VALUE FOR NUMBER OF THEATERS ',12)
000133 007 1127 FORMAT(.....ERROR.....ILLEGAL VALUE FOR NUMBER OF THEATER SETS ',12)
000134 008 IF(THRID(1) .GT.0 .AND. THRID(1) .LE. NTH) GO TO 1125
000135 006 PRINT 1124, THRID(1)
000136 006 1124 FORMAT(.....ERROR.....ILLEGAL VALUE FOR SET ',12)
000137 006 1125 IF(1 .GT. NTH) GO TO 1130
000138 006 IF(PLANID(1) .NE. ' ') GO TO 1130
000139 006 IER = 1
000140 006 PRINT 1126, I
000141 006 1126 FORMAT(.....ERROR.....BLANK PLANID FOR SET ',12)
000142 006 1130 CONTINUE
000143 006 IF(IER .NE. 0) RETURN 0
000144 006 C CONVERT CONSUMPTION AND ASK DATA TO TONS
000145 006 DO 7 I = 1,210
000146 006 ATON(I) = XTON(11/2000.
000147 006 DO 8 K = 1,NTH
000148 006 C SET FIRST MODE
000149 006 MOE(K) = MD(1)
000150 006 C FIX CDAY TO GET ALL UNITS ON THAT DAY
000151 006 8 CDAY(K) = CDAY(K) + 1
000152 006 C SET LAST DAY OF THEATER
000153 006 DO 9 K = 1,NTH
000154 006 DO 92 J = 1,8
000155 006 IF(PRD(J,K) .EQ. MZERO) GO TO 91
000156 006 92 CONTINUE
000157 006 LDAY(K) = PRD(8,K)
000158 006 GO TO 9
000159 006 91 LDAY(K) = PRD(J-1,K)
000160 006 9 CONTINUE
000161 006 C
000162 006 C WRITE HEADING
000163 006 PRINT 10
000164 006 10 FORMAT(1H,' DAY THR UNITS STRENGTH REPLACMNTS TYPE ',
000165 006 * 'DAYS ON DAYS DESRD TONS ON TONS REGRD')
000166 006 C
000167 006 C SORT MDF2 BY RDD
000168 006 CALL SOPEN3(850,870,63,10,KEY)
000169 006 50 READ(26,END=60) MDF2

```

UNCLASSIFIED

MRG PROGRAM NUR

```

000170      CALL SRREL(MDF2,63)
000171      GO TO 50
000172      60      CALL SSORT
000173      C
000174      C FIRST RECORD SHOULD BE SEQ NOS.
000175      70      CALL SRREL(MDF2,63,$1000)
000176      DO 90 I = 1,7
000177      90      SEQ(I) = MDF2(I)
000178      C
000179      C WRITE CARDS FOR THEATER BASE
000180      AVL = 999
000181      RDD = 0
000182      DO 96 K = 1,NTH
000183      DO 95 I = 1,3
000184      IFXRS(I,K) = EQ. 0.) GO TO 95
000185      SEQ(K) = SEQ(K) + 1
000186      WRITE(24,497) SEQ(K),PLANID(K),RT(1),RU(1),RN(1,1*5),RN(2,1*5),
000187      * LVL(1),MD(4),AVL,RDD,DEST(1,K),DEST(1,K),IZERO
000188      IF(1.EQ. 3) GO TO 94
000189      AMT = XRS(1,K) + .5
000190      WRITE(24,498) SEQ(K),PLANID(K),AMT
000191      GO TO 95
000192      94      AMT = XRS(1,K) / 1000. + .5
000193      WRITE(24,499) SEQ(K),PLANID(K),AMT
000194      C CONVERT BARRELS TO TONS
000195      XRS(1,K) = XRS(1,K) / 7.213
000196      95      CONTINUE
000197      96      CONTINUE
000198      C
000199      GO TO 399
000200      C MAIN LOOP
000201      C
000202      200      DAY=DAY+1
000203      C THEND = 0 - THEATER IS CURRENT
000204      C THEND = 1 - THEATER HAS ENDED. NEEDS DATA WRITTEN
000205      C THEND = 2 - THEATER HAS ENDED. ALL DATA WRITTEN
000206      DO 201 K = 1,NTH
000207      IF(THENDK) .NE. 0) GO TO 201
000208      PR = PER(K)
000209      IF(DAY .GT. PRO(PR,K)) PER(K) = PER(K) + 1
000210      IF(DAY .LE. LDAY(K)) GO TO 201
000211      PER(K) = PER(K) - 1
000212      THEND(K) = 1
000213      CONTINUE
000214      201      CONTINUE
000215      DO 202 K = 1,NTH
000216      IF(THENDK) .NE. 2) GO TO 206
000217      CONTINUE
000218      GO TO 900
000219      C
000220      C COMPUTE FILLER IF IT'S TIME
000221      DO 207 K = 1,NTH
000222      IF(DAY .NE. CDAY(K)) GO TO 207
000223      XFR(K) = XFR(K) + XSTREN(K)
000224      PRINT 410, DAY,K,XFR(K),XSTREN(K)
000225      LNS = LNS + 1
000226      207      CONTINUE
000227      C SET ANCHOR POINT FOR BUILDUP FUNCTION

```

UNCLASSIFIED

MRG PROGRAM NUR


```

000227 006 209 DO 220 K = 1,NTH
000228 006 IF(DAY*NE*RDAY(K)) GO TO 420
000229 009 IF(KU(K) *EQ. 0) GO TO 215
000230 006 IF(THEND(K)*NE. 0) GO TO 420
000231 006 PR = PER(K)
000232 006 DO 210 I=1,3
000233 006 C DAYS OF SUPPLY ON HAND AT END OF RDAY=1
000234 006 XRLEV(I,K) = XRS(I,K)/(XCNSH(I,K,PR)*XSTREN(K))
000235 006 XSLOPE(I,K) = (XLEVEL(I,K) - XRLEV(I,K))/(BLDUP(K) - RDAY(K))
000236 006 C ALREADY OVER REQUIRED LEVEL?
000237 006 IF(XSLOPE(I,K) *GT. 0.) GO TO 210
000238 006 XSLOPE(I,K) = 0
000239 006 XRLEV(I,K) = XLEVEL(I,K)
000240 006 210 CONTINUE
000241 009 GO TO 220
000242 009 C NO UNITS ON RDAY. UP IT TO NEXT DAY
000243 009 215 RDAY(K) = RDAY(K) + 1
000244 006 220 CONTINUE
000245 006 C
000246 006 300 CONTINUE
000247 006 IF(RDD*GT*DAY) GO TO 400
000248 006 C
000249 006 C
000250 006 C ENTER NEXT FORCE UNIT DATA
000251 006 DO 301 ITH = 1,IN
000252 006 IF(THR *EQ. THTR(ITH)) GO TO 302
000253 006 301 CONTINUE
000254 006 PRINT 2000, THTR, SRC, UIC, IPSN, THR
000255 006 2000 FORMAT(7I6, ' NO THEATER MATCH FOR ', 2A6, 1X, A6, 1X, 15, 1X, 13)
000256 006 LNS = LNS + 2
000257 006 GO TO 399
000258 006 302 ITH = THTRID(ITH)
000259 006 IF(THEND(ITH) *NE. 0) GO TO 399
000260 006 XPAX = PAX
000261 006 XSTREN(ITH)=XSTREN(ITH)+XPAX
000262 006 IF(KU(ITH) *EQ. 0) FDAY(ITH) = DAY
000263 006 KU(ITH)=KU(ITH)+1
000264 006 IF(DAY *LE. CDAY(ITH)) GO TO 399
000265 006 POM = 1
000266 006 IF(MCODE *EQ. 1) POM = 2
000267 006 C
000268 006 C ACCOMPANYING SUPPLY
000269 006 DO 305 I=1,3
000270 006 C TOTAL TONS ON HAND
000271 006 305 XRS(I,ITH)=XRS(I,ITH)+XASR(I,ITH,POM)*XPAX
000272 006 C
000273 006 C
000274 006 C READ NEXT UNIT
000275 006 399 CALL SRRET(MDF2,63,$1000)
000276 006 C
000277 006 GO TO 300
000278 006 C
000279 006 C
000280 006 C
000281 006 C CHECK 5-DAY INTERVAL
000282 006 400 IF(DAY/5*NE*DAY) GO TO 600
000283 006 C

```



```

000284      JDAY = DAY - 4
000285      C WRITE 5-DAY ROLLUPS
000286      C
000287      C CYCLE ON THEATER
000288      DO 520 K = 1,NTH
000289      IF(THEND(K) .EQ. 2) GO TO 520
000290      IF(THEND(K) .EQ. 1) THEND(K) = 2
000291      C
000292      C CHECK FOR DAY TO COMMENCE RESUPPLY
000293      IF(DAY.LT.RDAY(K)) GO TO 520
000294      IDAY = JDAY
000295      IF(IDAY .LT. FDAY(K)) IDAY = FDAY(K)
000296      IF(IDAY .GT. ADAY(K)) MOE(K) = MD(2)
000297      AVL = AVDAY(K)
000298      IF(AVL .LT. 0) AVL = IDAY + AVL
000299      IF(AVL .GE. IDAY) AVL = CDAY(K)
000300      C
000301      SMOB(1) = IDAY
000302      SMOB(2) = PLANID(K)
000303      SMOB(3) = DEST(2,K)
000304      SMOB(7) = MOE(K)
000305      SMOB(8) = AVL
000306      PR = PER(K)
000307      XDAY = DAY - RDAY(K)
000308      C
000309      C CYCLE ON SUPPLY TYPE
000310      C
000311      C
000312      IF(LNS .LT. 50) GO TO 403
000313      PRINT 10
000314      LNS = 1
000315      DO 510 I = 1,3
000316      C SET RESUPPLY REQUIREMENTS FOR TYPE
000317      C XDESIR = NEEDED - DAYS OF SUPPLY
000318      XDESIR = XSLOPE(1,K) * XDAY + XRLEV(1,K)
000319      IF(DAY .EQ. RDAY(K)) XDESIR = XRLEV(1,K)
000320      IF(DAY .GE. BLDUP(K)) XDESIR = XLEVEL(1,K)
000321      C XDESRO = NEEDED QUANTITY OF SUPPLY
000322      XDESRO = XDESIR * XCNSM(1,K,PR) * XSTREN(K)
000323      XAMT = XDESRO - XRS(1,K)
000324      IF(XAMT .LE. 0) GO TO 510
000325      XOH0 = XRS(1,K) / (XCNSM(1,K,PR) * XSTREN(K))
000326      C CONVERT TO THOUSANDS OF BARRELS
000327      IF(1 .EQ. 3) XAMT = 7.213 * XAMT / 1000.
000328      XRS(1,K) = XDESRO
000329      LNS = LNS + 1
000330      PRINT 410, DAY, K, KU(K), XSTREN(K), XAMPLCE(K), 1, XOH0, XDESIR, XRS(1,K),
000331      * XAMT
000332      410 FORMAT(1X,13,3X,11,3X,15,2(2X,F10.0),5X,11,2X,4(2X,F10.0))
000333      C
000334      C
000335      IST = 1
000336      IND = 18
000337      IF(1 .EQ. 3) IST = 19
000338      IF(1 .EQ. 3) IND = 19
000339      C CYCLE ON ORIGIN
000340      DO 500 J = 1ST,IND
000341      C

```

UNCLASSIFIED

```
000341 006 IF(I *EQ* 3) GO TO 450
000342 006 IF(XPCT(I,J) *EQ* 0) GO TO 500
000343 006 AMT = XAMT * XPCT(I,J) + .5
000344 006 GO TO 460
000345 006 AMT = XAMT + .5
000346 006 C
000347 006 AMT1 = AMT - 99999
000348 006 IF(AMT1 *GT* 0) AMT = 99999
000349 006 IF(AMT *GT* 12 *OR* I *EQ* 3) GO TO 480
000350 006 C SEND MINIMUM OF 12 SHORT TONS
000351 006 XRS(I,K) = XRS(I,K) + FLOAT(12 - AMT)
000352 006 AMT = 12
000353 006 C
000354 006 C WRITE MORSA CARD5
000355 006 SEQ(K) = SEQ(K) + 1
000356 006 WRITE(24,497) SEQ(K),PLANID(K),RT(I),RU(I),RN(1,I),RN(2,I),LVL(I),
000357 006 * MOE(K),AVL,1DAY,GORG(J),DEST(1,K),DEST(1,K),1ZERO
000358 006 * 2A1,2A6,A3,A1,2A3,J3,6X,3A4,J6,19X)
000359 006 C BULK/AMMO
000360 006 IF(I *NE* 3) WRITE(24,498) SEQ(K),PLANID(K),AMT
000361 006 FORMAT(J4,'A',A1,'W',6X,J6,61A)
000362 006 C POL
000363 006 IF(I *EQ* 3) WRITE(24,499) SEQ(K),PLANID(K),AMT
000364 006 FORMAT(J4,'A',A1,'W',12X,J6,55X)
000365 006 KM=KM+2
000366 006 SMOB(4) = 1
000367 006 SMOB(5) = AMT
000368 006 SMOB(6) = NORG(J)
000369 006 WRITE(23) SMOB
000370 006 WRITE(12,76) AVL,RT(I),RU(I),SEQ(K),GORG(J),DEST(1,K),MOE(K),1DAY
000371 006 * ,DEST(2,K),NORG(J),LVL(I),RN(1,I),RN(2,I),AMT,1
000372 006 FORMAT(3A,J3,2A1,4X,5A,J6,24A,A1,4X,1X,3J3,2X,5X,A3,2A6,4X,2J6)
000373 006 IF(AMT1 *LE* 0) GO TO 500
000374 006 AMT = AMT1
000375 006 GO TO 460
000376 006 500 CONTINUE
000377 006 510 CONTINUE
000378 006 C
000379 006 C REPLACEMENTS
000380 006 C
000381 006 C ENTER NOPAX
000382 006 IRPL = XMPLE(K) + .5
000383 006 IF(IRPL *EQ* 0) GO TO 520
000384 006 SEQ(K) = SEQ(K) + 1
000385 006 XMPLE(K) = 0.
000386 006 C
000387 006 WRITE(24,497) SEQ(K),PLANID(K),RT(I),RU(I),RN(1,I),RN(2,I),LVL(I),
000388 006 * MD(3),AVL,1DAY,GORG(20),DEST(1,K),DEST(1,K),IRPL
000389 006 KM=KM+1
000390 006 SMOB(4) = 4
000391 006 SMOB(5) = IRPL
000392 006 SMOB(6) = NORG(20)
000393 006 SMOB(7) = MD(3)
000394 006 WRITE(23) SMOB
000395 006 WRITE(12,496) AVL,RT(I),RU(I),SEQ(K),GORG(20),DEST(1,K),MD(3),1DAY
000396 006 * ,DEST(2,K),NORG(20),LVL(I),RN(1,I),RN(2,I),IRPL,SMOB(4)
000397 006 520 CONTINUE
```

UNCLASSIFIED


```

000455      * MOE(K),AVL,DAY,GORG(J),DEST(1,K),DEST(1,K),IZERO
000456      C BULK/AMMO
000457      IF(1.NE.3) WRITE(24,498) SEQ(K),PLANID(K),AMT
000458      C POL
000459      IF(1.EQ.3) WRITE(24,499) SEQ(K),PLANID(K),AMT
000460      KHEKM+2
000461      N = 1
000462      SMOB(4) = 1
000463      SMOB(5) = AMT
000464      SMOB(6) = NORG(J)
000465      WRITE(23) SMOB
000466      WRITE(12,496) AVL,RT(1),RD(1),SEQ(K),GORG(J),DEST(1,K),MOE(K),DAY,
000467      * DEST(2,K),NORG(J),LVL(1),RN(1,1),RN(2,1),AMT,1
000468      C30 CONTINUE
000469      C50 CONTINUE
000470      C CHECK FOR FILLER
000471      IF(DAY.NE.BDAY(1,K)) GO TO 690
000472      C SEND FILLER
000473      FIL = XFR(K) * XBER(1,K) + .5
000474      BDAY(1,K) = BDAY(2,K)
000475      BDAY(2,K) = BDAY(3,K)
000476      BDAY(3,K) = 9999
000477      XBER(1,K) = XBER(2,K)
000478      XBER(2,K) = XBER(3,K)
000479      SEQ(K) = SEQ(K) + 1
000480      WRITE(24,497) SEQ(K),PLANID(K),RT(1),RD(1),RN(1,5),RN(2,5),LVL(1),
000481      * MD(3),AVL,DAY,GORG(20),DEST(1,K),DEST(1,K),FIL
000482      KM = KM + 1
000483      SMOB(7) = MD(3)
000484      SMOB(4) = 4
000485      SMOB(5) = FIL
000486      SMOB(6) = NORG(20)
000487      WRITE(23) SMOB
000488      WRITE(12,496) AVL,RT(1),RD(1),SEQ(K),GORG(20),DEST(1,K),MD(3),DAY,
000489      * DEST(2,K),NORG(20),LVL(1),RN(1,5),RN(2,5),FIL,SMOB(4)
000490      C REPLACEMENTS
000491      C CHECK FOR D-DAY
000492      IF(DAY.LT.DDAY(K)) GO TO 700
000493      XADD = XTREN(K)*XRPLC(PR,K)/1000.
000494      XRPLC(K) = XRPLC(K) + XADD
000495      C700 CONTINUE
000496      C
000497      GO TO 200
000498      C
000499      C
000500      C
000501      C
000502      C PRINT STATISTICS
000503      C900 CONTINUE
000504      ENDFILE 24
000505      ENDFILE 23
000506      ENDFILE 12
000507      REWIND 23
000508      IF(ENDM) GO TO 909
000509      PRINT 901
000510      C901 FORMAT(' UNITS NOT PROCESSED ')
000511      C905 CALL SRRET(MDF2,63,8909)

```



```

000514 006 PRINT 902, SRC,UIC,TPSN,THR,RDD
000515 006 902 FORMAT(1X,2A6,1X,A6,3(1X,15))
000516 006 GO TO 905
000517 006 909 CALL SMOBST
000518 006 PRINT 910,KU,KM,SEW
000519 006 910 FORMAT(1X,UNITS IN THEATER
000520 006 * , FINAL SEQUENCE NUMBERS ', 717)
000521 006 STOP
000522 006 C
000523 006 C RAN OUT OF MDF2 BEFORE ALL PERIODS SATISFIED
000524 006 1000 PRINT 1010, DAY
000525 006 1010 FORMAT(1X,END OF MDF2 FILE, DAY = ',15)
000526 006 RDD = 999999
000527 006 ENDM = .TRUE.
000528 006 GO TO 400
000529 006 END

```

END ELT.

MRG PROGRAM MEGEMORSA ••UNCLASSIFIED•• .L.O

```
WELT.L 58CAMP.MEGEMORSA
ELT007 RL1870 08/16-07:43:47-(0.)
000001 000 IMPLICIT INTEGER (A-Z)
000002 000 DIMENSION IN(14),SEQ(14),SVC(14),PLAN(14),INN(13)
000003 000 EQUIVALENCE (IN(2),INN(1))
000004 000 DATA SVC/14,'A'/
000005 000 C MERGES UNIT AND NON-UNIT MORSA CARDS ON ONE FILE
000006 000 IGO = 1
000007 000 C UNIT CARDS
000008 000 10 READ(10,1,END=19) IPLN,INN
000009 000 1 FORMAT(5X,A1,12A6,A2)
000010 000 GO TO 100
000011 000 C READ NON-UNIT CARDS
000012 000 19 IGO = 2
000013 000 20 READ(24,1,END=200) IPLN,INN
000014 000 C FIX UP SEQ NO. IF NECESSARY
000015 000 C CARD TYPE A?
000016 000 100 IF (FLD(10,0,IN(2)) .NE. '000000A') GO TO 140
000017 000 DO 110 I = 1,14
000018 000 IF (PLAN(I) .EQ. IPLN) GO TO 130
000019 000 IF (PLAN(I) .EQ. 0) GO TO 125
000020 000 CONTINUE
000021 000 PLAN(I) = IPLN
000022 000 SEQ(I) = SEQ(I) + 1
000023 000 IF (SEQ(I) .LT. 10000) GO TO 140
000024 000 C OVER 4 CHAR, START OVER WITH SVC CODE = B
000025 000 SEQ(I) = 1
000026 000 SVC(I) = 'B'
000027 000 C FIX UP WORD ONE
000028 000 140 ENCODE(6,150,IN(1)) SEQ(1),SVC(1),IPLN
000029 000 150 FORMAT(J4,A1,A1)
000030 000 CALL NTRAN(8,1,14,IN,L,22)
000031 000 GO TO 110,20), IGO
000032 000 C WRITE END OF FILE
000033 000 200 CALL NTRAN(8,9)
000034 000 PRINT 210,PLAN,SVC,SEQ
000035 000 210 * ,3X,14(J4,12X)
000036 000 FORMAT(1X,'PLANID',5X,14A6/1X,'SERVICE',4X,14A6/1X,'SEQUENCE',
000037 000 * ,3X,14(J4,12X))
000038 000 STOP
000039 000 END
```

END ELT.

UNCLASSIFIED

MRG PROGRAM PKAGE

BHDG

WELT.L 58CAMP.PCKAGE

ELT007 RL1870 08/16-07:43:49-(15,)

000001 C CREATS SMOBSMOD PACKAGES OF MDF2 AND NUR OUTPUT

000002 C INPUT FILES 26 - MDF2, 23 - SMOB OUTPUT FROM NUR (SORTED)

000003 C OUTPUT FILES 22 - SMOBSMOD CARDS

000004 C IMPLICIT INTEGER (A-Y)

000005 C DIMENSION MDF2(63),KEY(31)

000006 C DIMENSION SRC(2),MDF1(16)

000007 C EQUIVALENCE (MDF1(1),SRC), @ 246

000008 C (MDF1(3),UIC), @ A6

000009 C (MDF1(4),TPSN), @ 15

000010 C (MDF1(8),ONOD), @ 13

000011 C (MDF1(9),DNOD), @ 13

000012 C (MDF1(10),RDO), @ 13

000013 C (MDF1(11),AVL), @ 13

000014 C (MDF1(12),MODE), @ A1

000015 C (MDF1(13),THR), @ 11

000016 C (MDF1(14),MCOE), @ 11

000017 C DIMENSION TUCHA(22,2)

000018 C EQUIVALENCE (TUCHA(5,2), PAX),

000019 C (TUCHA(6,1),ZVEHNA),

000020 C (TUCHA(7,1),ZNSDNA), (TUCHA(7,2),NSDNA),

000021 C (TUCHA(8,1),ZNSDOU), (TUCHA(8,2),NSDOU),

000022 C (TUCHA(9,1),ZVEHOU),

000023 C (TUCHA(10,1),ZNVOUT),

000024 C (TUCHA(11,1),ZNSOVV), (TUCHA(11,2),NSDOVVC),

000025 C (TUCHA(12,1),ZNSOVN), (TUCHA(12,2),NSDOVNV),

000026 C (TUCHA(13,1),ZVHOVC),

000027 C (TUCHA(14,1),ZVHOVN),

000028 C (TUCHA(15,1),ZNVOVN),

000029 C (TUCHA(16,1),ZNVOVV),

000030 C (TUCHA(17,1),ZVEHBC),

000031 C (TUCHA(18,1), ZNVBC)

000032 C EQUIVALENCE (MDF2,MDF1), (MDF2(18),TUCHA(1,1)),

000033 C EQUIVALENCE (MDF2(62),ASUPPLY), (MDF2(63),ACAMMO)

000034 C DATA KEY/ 9,1,36,1,0,1, @ UNOD

000035 C 12,1,36,1,0,2, @ MODE (MODIFIED)

000036 C 8,1,36,1,0,3, @ ONOD

000037 C 10,1,36,1,0,4, @ RUD

000038 C 11,1,36,1,0,5, @ AVL

000039 C 99999/

000040 C LOGICAL ENDM, ENDS

000041 C DIMENSION SMOB(8)

000042 C 1 = RDD, 2 = PLANID, 3 = DEST, 4 = TYPE, 5 = AMT, 6 = ORIG, 7 = MODE,

000043 C 8 = AVL

000044 C DIMENSION ZTUCHA(22,2)

000045 C EQUIVALENCE (TUCHA,ZTUCHA)

000046 C DIMENSION STOT(17,3),ZSTOT(17,3),MTOT(17),ZMTOT(17)

000047 C EQUIVALENCE (STOT,ZSTOT), (MTOT,ZMTOT)

000048 C DIMENSION IN(24),ZIN(24),KEY1(37)

000049 C EQUIVALENCE (IN,ZIN)

000050 C DATA KEY1/ 2,1,36,1,0,1, @ RDD

000051 C 3,1,36,1,0,2, @ AVL

000052 C 4,1,36,1,0,3, @ DEST

000053 C 5,1,36,1,0,4, @ ORIG

000054 C 1,1,36,1,0,5, @ PCK

000055 C 6,1,36,1,0,6, @ MODE

```

000056 001 * 99999/
000057 001 DIMENSION PACK(4,600),KPKC(3), MD1(4),MD2(4), MINTP(2)
000058 001 DATA MD1 /1,0,2,3/, MD2 /2,1,3,4/, MINTP(2)/100000/
000059 003 DIMENSION XTHR(7), XDAY(7)
000060 001 C
000061 001 C
000062 001 WRITE(21,1)
000063 001 1 FORMAT(5X,'SRC',9X,'UIC',4X,'TPSN PACK DEST ORIG MODE '
000064 001 * 'RDD AVL UNITS TOT PAX TOT STON TOT SQ FT')
000065 001 WRITE(20,2)
000066 001 2 FORMAT(' PACK DEST ORIG MODE RDD AVL = UNIT',
000067 001 * ' TOT PAX TOT STON SQ FT/BRL')
000068 001 C
000069 001 C SORT MDF2
000070 001 C
000071 001 CALL SOPENJ(10,850,63,12,KEY)
000072 001 10 READ(26,END=30) MDF2
000073 001 C SKIP IF MORSA SEQ NUMBERS
000074 001 IF(RDD.EQ.-99999) GO TO 10
000075 001 C SKIP IF NO MOVE
000076 001 IF(MODE.EQ.'S') M = 4
000077 001 C IF(MODE.EQ.'A') M = 1
000078 001 IF(MODE.EQ.'S') M = 4
000079 001 IF(MODE.EQ.'A') M = 1
000080 001 IF(MODE.EQ.'S') M = 4
000081 001 IF(MODE.EQ.'A') M = 1
000082 001 IF(MODE.EQ.'S') M = 4
000083 001 IF(MODE.EQ.'A') M = 1
000084 001 IF(MODE.EQ.'S') M = 4
000085 001 IF(MODE.EQ.'A') M = 1
000086 001 IF(MODE.EQ.'S') M = 4
000087 001 IF(MODE.EQ.'A') M = 1
000088 001 IF(MODE.EQ.'S') M = 4
000089 001 IF(MODE.EQ.'A') M = 1
000090 001 IF(MODE.EQ.'S') M = 4
000091 001 IF(MODE.EQ.'A') M = 1
000092 001 IF(MODE.EQ.'S') M = 4
000093 001 IF(MODE.EQ.'A') M = 1
000094 001 IF(MODE.EQ.'S') M = 4
000095 001 IF(MODE.EQ.'A') M = 1
000096 001 IF(MODE.EQ.'S') M = 4
000097 001 IF(MODE.EQ.'A') M = 1
000098 001 IF(MODE.EQ.'S') M = 4
000099 001 IF(MODE.EQ.'A') M = 1
000100 001 IF(MODE.EQ.'S') M = 4
000101 001 IF(MODE.EQ.'A') M = 1
000102 001 IF(MODE.EQ.'S') M = 4
000103 001 IF(MODE.EQ.'A') M = 1
000104 001 IF(MODE.EQ.'S') M = 4
000105 001 IF(MODE.EQ.'A') M = 1
000106 001 IF(MODE.EQ.'S') M = 4
000107 001 IF(MODE.EQ.'A') M = 1
000108 001 IF(MODE.EQ.'S') M = 4
000109 001 IF(MODE.EQ.'A') M = 1
000110 001 IF(MODE.EQ.'S') M = 4
000111 001 IF(MODE.EQ.'A') M = 1
000112 001 IF(MODE.EQ.'S') M = 4

```



```

000170 001 C WRITE OUT PACKAGE - UNIT DATA
000171 001 ZAVL = AVLSAV
000172 001 PCK = PCK + 1
000173 001 ZTOTST = 0
000174 001 ZTOTSF = 0
000175 001 DO 260 I = 1,14
000176 001 GO TO (255,251,255,255,251,251,255,255,251,251,251,251,251,251,251),I
000177 001 251 ZTOTOT(1) = ZTOTOT(1) / 1000.
000178 001 ZTOTST = ZTOTST + ZTOTOT(1)
000179 001 GO TO 260
000180 001 ZTOTOT(1) = FLOAT(ZTOTOT(1)) / 1000.
000181 001 IF(I.EQ.1) GO TO 260
000182 001 ZTOTSF = ZTOTSF + ZTOTOT(1)
000183 001 260 CONTINUE
000184 001 DO 270 I = 15,7,-1
000185 001 ZTOTOT(1) = ZTOTOT(1)-1
000186 001 ZTOTOT(6) = 0
000187 001 C CHANGE PACKAGE MODE IF NECESSARY
000188 003 DO 271 I = 1,7
000189 004 IF(THRSV.EQ.XTHR(1)) GO TO 273
000190 003 271 CONTINUE
000191 003 PRINT 272, THRSV
000192 003 272 FORMAT(' ERROR IN THEATER INPUT. NEED ENTRY FOR THEATER',15)
000193 003 RETURN 0
000194 003 273 IF(RDDSAV.LT.XDAY(1)) GO TO 279
000195 003 M12 = 4
000196 003 WRITE(22) PCK,ZRDU,ZAVL,M9,M8,M12,(ZTOTOT(1),I=1,17),MINTP(CBT)
000197 001 C ZERO PACKAGE DATA
000198 001 IF(NPCK.GT.MAX) MAX = NPCK
000199 001 WRITE(21,281) ((PACK(I,J),I=1,4),PCK,M9,M8,M12),ZRDU,
000200 001 * ZAVL,NPCK,ZTOTOT(1),ZTOTST,ZTOTSF),J=1,NPCK)
000201 001 281 FORMAT(11X,2A6,2X,A6,5(2X,15),2(2X,F4,U),2X,15,3(2X,F9,3))
000202 001 WRITE(20,422) PCK,M9,M8,M12),ZRDU,ZAVL,NPCK,ZTOTOT(1),ZTOTST,
000203 001 * ZTOTSF,MINTP(CBT)
000204 001 NPCK = 0
000205 001 DO 280 I = 1,14
000206 001 280 ZTOTOT(1) = 0
000207 001 IF(IGO.EQ.0) GO TO 299
000208 001 IGO = 0
000209 001 MINTP(1) = TPSN
000210 001 IF(JGO.EQ.0) GO TO 150
000211 001 JGO = 0
000212 001 MINTP(1) = 100000
000213 001 GO TO 199
000214 001 299 IF(.NOT.ENDM) GO TO 200
000215 001 C READ FIRST SMOB RECORD
000216 001 HEAD(23) SMOB
000217 001 KV = 1
000218 001 GO TO 510
000219 001 C
000220 001 C READ NUR DATA
000221 001 300 IF(SMOB(6).NE.S6.OR.SMOB(3).NE.S3
000222 001 *OR.SMOB(1).GT.LIMRDU.OR.SMOB(8).GT.LIMAVL) GO TO 400
000223 001 C ADD TO PACKAGE - NUR DATA
000224 001 C 1 = DRY BULK (TONS), 2 = AMMO (TONS), 3 = POL (1000 BARRELS), 4 = PEOPLE
000225 001 C OUTPUT ALL IN THOUSANDS
000226 001 C

```

```
000227 001 360 M = 2
000228 001 IF(SMOB(7) .EQ. 'A') M = 1
000229 001 IF(SMOB(7) .EQ. 'S') M = 3
000230 001 IF(SMOB(8) .GT. AVLSAV) AVLSAV = SMOB(8)
000231 001 I = SMOB(4)
000232 001 STOT(1,M) = STOT(1,M) + SMOB(5)
000233 001 IF(1 .EQ. 3 .AND. STOT(1,M) .GT. 99999) GO TO 362
000234 001 IF(STOT(1,M) .LE. 99999999) GO TO 370
000235 001 C
000236 001 C SUBTRACT FROM OVERFILLED PACKAGE
000237 001 362 STOT(1,M) = STOT(1,M) - SMOB(5)
000238 001 IGO = M
000239 001 PRINT 361, 1,M,STOT(1,M),SMOB(5),56,53
000240 001 FORMAT(' SMOB ',6110)
000241 001 GO TO 400
000242 001 C
000243 001 370 KPCK(M) = KPCK(M) + 1
000244 001 C
000245 001 READ(23,END=550) SMOB
000246 001 KV = KV + 1
000247 001 GO TO 300
000248 001 C
000249 001 400 CONTINUE
000250 001 C WRITE OUT PACKAGE = NUR DATA
000251 001 ZAVL = AVLSAV
000252 001 IS = 1
000253 001 IE = 3
000254 001 IF(1GO .NE. 0) IS = IGO
000255 001 IF(1GO .NE. 0) IE = IGO
000256 001 DO 435 M = JS,JE
000257 001 IF(KPCK(M) .EQ. 0) GO TO 435
000258 001 PCK = PCK + 1
000259 001 MPCK = MPCK - 1
000260 001 ZSTOT(6,M) = STOT(3,M)
000261 001 ZSTOT(16,M) = FLOAT(STOT(1,M))/1000.
000262 001 ZSTOT(17,M) = FLOAT(STOT(2,M))/1000.
000263 001 ZSTOT(1,M) = FLOAT(STOT(4,M))/1000.
000264 001 ZTOTST = ZSTOT(16,M) + ZSTOT(17,M)
000265 001 C ZERO PACKAGE DATA
000266 001 DO 430 I = 2,4
000267 001 STOT(1,M) = 0
000268 001 WRITE(22) PCK,ZRDU,ZAVL,53,56,M,(ZSTOT(1,M),I=1,17),MPCK
000269 001 WRITE(20,422) PCK,53,56,M,1(M),ZRDU,ZAVL,KPCK(M),ZSTOT(1,M),
000270 001 * ZTOTST,ZSTOT(6,M),MPCK
000271 001 WRITE(19,422) PCK,53,56,M,1(M),ZRDU,ZAVL,KPCK(M),ZSTOT(1,M),
000272 001 * ZTOTST,ZSTOT(6,M)
000273 001 422 FORMAT(4(2X,15),2(2X,F4.0),2X,15,3(2X,F9.3),2X,15)
000274 001 KPCK(M) = 0
000275 001 STOT(1,M) = 0
000276 001 CONTINUE
000277 001 IF(1GO .EQ. 0) GO TO 440
000278 001 IGO = 0
000279 001 GO TO 360
000280 001 IF(ENDS) GO TO 1000
000281 001 GO TO 510
000282 001 C
000283 001 500 ENDM = .TRUE.
```



```
000284 001 GO TO 250
000285 001 C
000286 001 C SAVE DATA FROM PACKAGE'S STARTING RECORD
000287 001 510 S6 = SMOB(6)
000288 001 S3 = SMOB(3)
000289 001 ZRDD = SMOB(1)
000290 001 LIMRDD = SMOB(1) + DTRDD
000291 001 LIMAVL = SMOB(8) + DTAVL
000292 001 AVLSAV = SMOB(8)
000293 001 GO TO 360
000294 001 550 ENDS = .TRUE.
000295 001 GO TO 400
000296 001 C
000297 001 1000 ENDFILE 22
000298 001 REWIND 22
000299 001 C
000300 001 C SORT 22 BY AVL, RDD
000301 001 CALL SOPEN3(1005,1030,24,6,KEY1)
000302 001 C READ OTHER SERVICE DATA, THEN FILE 22
000303 001 1005 READ(18,1006,END=1010) IN
000304 001 1006 FORMAT(16,2F3.0,2I3,11,6(F8.3)/11(F8.3),16)
000305 001 C FIX UP MODE FOR SORT
000306 001 J = IN(6) + 1
000307 001 IN(6) = MD2(IJ)
000308 001 K0 = K0 + 1
000309 001 PCK = PCK + 1
000310 001 CALL SRRELTIN(24)
000311 001 GO TO 1005
000312 001 1010 READ(22,END=1020) IN
000313 001 CALL SRRELTIN(24)
000314 001 GO TO 1010
000315 001 1020 REWIND 22
000316 001 CALL SSORT
000317 001 1030 CALL SRRELTIN(24,1100)
000318 001 SEW = SEW + 1
000319 001 C OTHER SERVICE DATA?
000320 001 IF(IN(24) .EQ. 999999) GO TO 1040
000321 001 TP = .
000322 001 IF(IN(24) .GT. 0 .AND. IN(24) .LT. 100000) ENCODE(5,1029,TP,L) IN(24)
000323 001 1029 FORMAT(J5)
000324 001 WRITE(22,1031) IN(1),TP,IN(2),IN(3),SEW
000325 001 1031 FORMAT(' PACKAGE ',J6,5X,A5,1X,'RUD= ',F6.2,6X,'AVL= ',F6.4,16)
000326 001 IF(IN(24) .LT. 0) GO TO 1050
000327 001 1035 5 = SEW
000328 001 GO TO 1060
000329 001 C OTHER SERVICE DATA
000330 001 1040 WRITE(22,1041) (IN(1),1=1,3),SEW
000331 001 1041 FORMAT(' OTHER SERVICE PACKAGE ',J6,7X,'RDD= ',F6.2,6X,
000332 001 ' AVL= ',F6.2,16)
000333 001 GO TO 1035
000334 001 1050 5 = -SEW
000335 001 C SET MODE PROPERLY FOR CARDS
000336 001 1060 J = IN(6)
000337 001 IN(6) = MD1(J)
000338 001 IF(IN(7) .NE. 0) WRITE(22,1032) (IN(1),1=3,7),5
000339 001 DO 1070 I = 8,23
000340 001 N = I - 6
```


MRG PROGRAM PKAGE

MRG PROGRAM PKAGE

```

000341      IF(IN(1) .NE. 0) WRITE(22,1033) (IN(K),K=3,6),ZIN(1),N,S
000342      1070 CONTINUE
000343      1032 FORMAT(1X,F9.2,2X,13.2X,13.4X,11.1X,F9.3,14X,'1',15)
000344      1033 FORMAT(1X,F9.2,2X,13.2X,13.4X,11.1X,F9.3,14X,12.15)
000345      1034 FORMAT(3X,'5444',22X,'0','8X','0.')
000346      WRITE(22,1034)
000347      GO TO 1030
000348      C
000349      1100 ENDFILE 22
000350      MPCK = -MPCK
000351      PRINT 1101, KU,KV,PCK,MPCK,KO,MAX
000352      1101 FORMAT(' UNITS PROCESSED ',17//
000353      * NUR PROCESSED ',17//
000354      * TOT # OF PKAGES',17//
000355      * # OF NUR PACKS ',17//
000356      * OTHER SERVICES ',17//
000357      * MAX UNITS IN PCK',17)
000358      STOP
000359      END

```

END ELT.

MRG POST-PROCESSOR PROGRAM INLAY

MRG POST-PROCESSOR PROGRAM INLAY

MRG POST-PROCESSOR PROGRAM INLAY

MRG POST-PROCESSOR PROGRAM INLAY

MRG PROGRAM PKAGE

MRG PROGRAM PKAGE

MRG POST-PROCESSOR PROGRAM INLAY

MRG POST-PROCESSOR PROGRAM INLAY

MRG POST-PROCESSOR PROGRAM INLAY

```

BELT-L 58CAMP-LAYIN
ELT007 RL1870 08/16-07:43:54--(25,)
000001 021 IMPLICIT INTEGER(A-Z)
000002 021 DIMENSION F(52), OF(52,24), @ UNITS 17,15 FAS INPUT, OUTPUT
000003 021 U(19), @ UNIT 11 LOG INPUT
000004 021 N(13), @ UNIT 12 NUR INPUT
000005 021 P( 2) @ UNIT 13 PACKAGE INPUT
000006 021 EQUIVALENCE (N,U), (OF,F)
000007 021 DIMENSION EDATE(2)
000008 021 DIMENSION KEYU(7), KEYP(7), KEYF(7), KEYOF(7)
000009 021 DATA KEYU /2, 1,36,0,0,1,99999/ @ UIC IN UNIT DATA
000010 021 DATA KEYP /1, 1,36,0,0,1,99999/ @ UIC IN PACKAGE DATA
000011 021 DATA KEYF /1,19,36,0,0,1,99999/ @ UIC IN FAS
000012 021 DATA KEYOF/1, 7,90,0,0,1,99999/ @ KEY FIELD IN OUTPUT FAS
000013 021 LOGICAL ENDO, ENDP
000014 021 DATA ZERO/'00000000'/
000015 021 DEFINE C01(X) = FLD(0,6,X)
000016 021 DEFINE C02(X) = FLD(6,6,X)
000017 021 DEFINE C03(X) = FLD(12,6,X)
000018 021 DEFINE C04(X) = FLD(18,6,X)
000019 021 DEFINE C05(X) = FLD(24,6,X)
000020 021 DEFINE C06(X) = FLD(30,6,X)
000021 021 DEFINE C12(X) = FLD(0,12,X)
000022 021 DEFINE C13(X) = FLD(0,18,X)
000023 021 DEFINE C14(X) = FLD(0,24,X)
000024 021 DEFINE C15(X) = FLD(0,30,X)
000025 021 DEFINE C23(X) = FLD(6,12,X)
000026 021 DEFINE C24(X) = FLD(6,18,X)
000027 021 DEFINE C25(X) = FLD(6,24,X)
000028 021 DEFINE C26(X) = FLD(6,30,X)
000029 021 DEFINE C34(X) = FLD(12,12,X)
000030 021 DEFINE C35(X) = FLD(12,18,X)
000031 021 DEFINE C36(X) = FLD(12,24,X)
000032 021 DEFINE C45(X) = FLD(18,12,X)
000033 021 DEFINE C46(X) = FLD(18,18,X)
000034 021 DEFINE C56(X) = FLD(24,12,X)
000035 021 C
000036 021 C SORT UNIT DATA
000037 021 C
000038 021 CALL SOPEN3(10,330,19,2,KEYU)
000039 021 10 READ(11,1,END=20) U
000040 021 1 FORMAT(19A6)
000041 021 CALL SRREL(U,19)
000042 021 KU = KU + 1
000043 021 GO TO 10
000044 021 REWIND 11
000045 021 CALL SSORT
000046 021 30 CALL SRRET(U,19,340)
000047 021 WRITE(11,1) U
000048 021 GO TO 30
000049 021 ENDFILE 11
000050 021 REWIND 11
000051 021 C SORT PACKAGE DATA
000052 021 C
000053 021 C
000054 021 C READ HEADER RECORD
000055 021 READ(21,51) P

```

```

000056 021 CALL SOPEN3(150,170,2,1,KEYP)
000057 021 C PICK UP UIC AND PACKAGE NUMBER
000058 021 50 READ(12,151,END=60) P
000059 021 51 FORMAT(15A,16,18X,A6)
000060 021 CALL SRREL(P,2)
000061 021 KP = KP + 1
000062 021 GO TO 50
000063 021 60 CALL SSORT
000064 021 70 CALL SRRET(P,2,180)
000065 021 WRITE(13,1) P
000066 021 GO TO 70
000067 021 80 ENDFILE 13
000068 021 REWIND 13
000069 021 C
000070 021 C SORT FAS FILE
000071 021 CALL SOPEN3(100,120,52,2,KEYF)
000072 021 100 READ(17,END=110) F
000073 021 CALL SRREL(F,52)
000074 021 KF = KF + 1
000075 021 GO TO 100
000076 021 110 CALL SSORT
000077 021 C NOW PROCESS UNIT RECORDS
000078 021 C
000079 021 120 CALL SRRET(F(1),52,155)
000080 021 UIC = C46(F(1))20,18 + C13(F(2))
000081 021 LF = LF + 1
000082 021 READ(11,1) U
000083 021 READ(13,1) P
000084 021 LU = LU + 1
000085 021 LP = LP + 1
000086 021 IF(U(2) - P(1)) 125,130,121
000087 021 121 PRINT 122, U(2),P(1)
000088 021 122 FORMAT(' NO MATCH FOR UNIT ',A6,' ON PACKAGE FILE ',A6)
000089 021 STOP
000090 021 PKC = ZERO
000091 021 GO TO 131
000092 021 130 PKC = P(2)
000093 021 READ(13,1) P
000094 021 LP = LP + 1
000095 021 131 IF(U(2) - UIC) 140,160,150
000096 021 C NO MATCH
000097 021 140 PRINT 141, U(2),UIC
000098 021 141 FORMAT(' NO MATCH FOR UNIT ',A6,' ON FAS ',A6/
000099 021 * , JOB IS BEING TERMINATED ')
000100 021 GO TO 515
000101 021 C NO MATCH, KEEP LOOKING
000102 021 150 WRITE(14) F
000103 021 CALL SRRET(F(1),52,155)
000104 021 LF = LF + 1
000105 021 IF(ENDU) GO TO 150
000106 021 UIC = C46(F(1))20,18 + C13(F(2))
000107 021 GO TO 131
000108 021 155 IF(ENDU) GO TO 180
000109 021 PRINT 156, U(2)
000110 021 156 FORMAT(' REACHED END OF FAS BEFORE UNITS ',A6)
000111 021 STOP
000112 021 C MATCHED FAS, ADD DATA

```

```

000113 025 C SAVE POSITIONS 37,38 IN 120,121
000114 025 160 C06(F(20)) = C01(F(7))
000115 025 C01(F(21)) = C02(F(7))
000116 025 C46(F( 6)) = U(1) * AVL
000117 021 C01(F( 7)) = '999999' * RT
000118 021 C02(F( 7)) = U(3) * RD
000119 021 C06(F( 8)) = U(4) * MORSA SEQ NO
000120 021 F( 9) = U(5) * MORSA SEQ NO, GEO ORIG
000121 021 F(10) = U(6) * GEO ORIG, DEST
000122 021 C12(F(11)) = C12(U(7)) * GEO DEST, MODE
000123 021 C26(F(19)) = U(8) * RDD, NODE DEST
000124 021 C14(F(20)) = C14(U(9)) * NODE DEST, ORIG
000125 023 C15(F(28)) = PCK * PACKAGE NO
000126 021 C06(F(38)) = U(10) * BULK
000127 021 DO 170 I = 1,18
000128 021 F(1+28) = U(1) * BULK,OVER,OUT,NAT,NSDA,TOTL,SUPPLY,PAX,AMMO
000129 021 C01(F(47)) = C01(U(19)) * AMMO
000130 021 C
000131 021 READ(11,1,END=175) U
000132 021 LU = LU + 1
000133 021 IF(ENDP) GO TO 171
000134 021 IF(U(2) - P(1)) 171,172,121
000135 021 171 PCK = ZERO
000136 021 GO TO 150
000137 021 172 PCK = P(4)
000138 021 READ(13,1,END=176) P
000139 021 LP = LP + 1
000140 021 GO TO 150
000141 021 175 ENDP = .TRUE.
000142 021 GO TO 150
000143 021 176 ENDP = .TRUE.
000144 021 GO TO 171
000145 021 C
000146 021 C SORT ACCORDING TO KEY FIELD
000147 021 180 ENDFILE 14
000148 021 REMIND 14
000149 021 CALL SOPEN(3,185,3195,52,3,KEYOF)
000150 021 185 READ(14,END=190) F
000151 021 CALL SRREL(F,52)
000152 021 GO TO 185
000153 021 190 CALL SSORT
000154 021 195 DO 198 I = 1,24
000155 021 ISAV = I
000156 021 CALL SRRET(OF(1,1),52,$200)
000157 021 CONTINUE
000158 021 CALL NTRAN(15,1,1248,OF,L,22)
000159 021 GO TO 195
000160 021 C
000161 021 C PROCESS NON UNIT DATA - GENERATE NEW FAS RECORDS
000162 021 C
000163 021 C READ INPUT DATA
000164 021 200 READ(5,211) FICOD,EDATE
000165 021 211 FORMAT(A1,X,A6,A1)
000166 021 PRINT 212, FICOD,EDATE
000167 021 212 FORMAT(' FICOD = ',A1,' EDATE = ',A6,A1)
000168 021 F1 = ' '
000169 021 F2 = ' '

```


••UNCLASSIFIED••

HRG POST-PROCESSOR PROGRAM INLAY

```

000170 021 F3 = ' '
000171 021 C04(F1) = C01(FICOD)
000172 021 C56(F1) = '888889N'
000173 021 C36(F2) = C13(EDATE(1))
000174 021 C13(F3) = EDATE(1)
000175 021 C04(F3) = C01(EDATE(2))
000176 021 ICT = ISAV - 1
000177 021 C
000178 021 220 READ(12,221,END=500) N,ITYP
000179 021 221 FORMAT(13A6,16)
000180 021 ICT = ICT + 1
000181 021 KN = KN + 1
000182 021 DO 225 I = 4,52
000183 021 OF(1,ICT) = ' '
000184 021 C
000185 021 C SET FAS DATA
000186 021 C
000187 021 OF(2,ICT) = F2
000188 021 OF(3,ICT) = F3
000189 021 OF(12,ICT) = ' ' DC*
000190 021 SEQ = SEQ + 1
000191 021 ENCODE(9,226,OF(1,ICT),L) SEQ
000192 021 226 FORMAT(4X,J5)
000193 021 C24(OF(1,ICT)) = F1
000194 021 DO 227 I = 1,6
000195 021 OF(1+5,ICT) = N(1) @ AVL,RT,RD,MORSA SEQ NO, GEO ORG,DEST,MODE
000196 021 DO 228 I = 7,8
000197 021 OF(1+12,ICT) = N(1) @ RCD,NUDE DEST,ORIG
000198 021 DO 229 I = 9,12
000199 021 OF(1+25,ICT) = N(1) @ LVL,NAME
000200 021 GO TO (231,233,233,234), I,ITP
000201 021 C DRY/AMMO
000202 021 231 C06(OF(38,ICT)) = C02(N(13)) @ AMT
000203 021 C14(OF(39,ICT)) = N(13) @ AMT
000204 021 C15(OF(43,ICT)) = N(13) @ TOTAL
000205 021 GO TO 240
000206 021 C POL
000207 021 233 C56(OF(44,ICT)) = C23(N(13)) @ AMT
000208 021 C13(OF(45,ICT)) = N(13) @ AMT
000209 021 GO TO 240
000210 021 C PAX
000211 022 234 C46(OF(45,ICT)) = C24(N(13)) @ AMT
000212 021 C12(OF(46,ICT)) = N(13) @ AMT
000213 021 C
000214 021 240 IF(ICT.NE.24) GO TO 220
000215 021 CALL NTRAN(15,1,1248,OF,L,22)
000216 021 ICT = 0
000217 021 GO TO 220
000218 021 C
000219 021 500 ICT = ICT + 1
000220 021 550 DO 560 J = 1CT,24
000221 021 DO 560 K = 1,52
000222 021 OF(K,J) = '999999'
000223 024 CALL NTRAN(15,1,1248,OF,L,9,10,22)
000224 021 515 PRINT 511, KU,LU,KP,LP,KF,LF,KN
000225 021 511 FORMAT(' UNITS READ ',16,' PROCESSED ',16/
000226 021 ' PACKAGED UNITS READ ',16,' PROCESSED ',16/

```

••UNCLASSIFIED••

HRG POST-PROCESSOR PROGRAM INLAY

••UNCLASSIFIED••

MRG POST-PROCESSOR PROGRAM INLAY

000227 021 • • FAS UNITS READ ' ,16,' PROCESSED ' ,16/
 000228 021 • • NON UNIT FAS RECORDS CREATED ' ,16)
 000229 021 STOP
 000230 021 END

END ELT.

000G MRG POST-PROCESSOR PROGRAM PRTPCK ••UNCLASSIFIED•• •L,0

••UNCLASSIFIED••

MRG POST-PROCESSOR PROGRAM INLAY

```

WELT,L 58CAMP,PRTPKC
ELT007 RL1870 08/16-07:43:57-(11,)
000001 007 C PRINT UNITS IN PACKAGES
000002 007 IMPLICIT INTEGER (A-Y)
000003 007 DIMENSION IN(11),ZIN(3),PRT(4,4),JN(7),ZUN(3)
000004 007 C READ HEADING RECORD
000005 007 READ(21,1) IN(1)
000006 007 PRINT 2
000007 007 2 FORMAT(1H1,' PACKAGE DEST ORIG MODE RDD AVL UNITS '
000008 007 * ,TOT PAX TOT STON TOT SQ FT ,22H1000'S OF PAX, TONS,
000009 007 * , SQ FT),// 10X,4(9X,'SKC',7X,'UIC',4X,'TPSN',)
000010 010 10 READ (21,1,END=80) IN,ZIN
000011 007 1 FORMAT(1X,24(2X,A6,5(2X,15)),2X,13,3X,13,3X,15,3(2X,F9.3))
000012 007 IF(IN(5) .EQ. PACK) GO TO 50
000013 007 IF(1CT .NE. 0) GO TO 70
000014 007 15 PACK = IN(5)
000015 007 1CT = 0
000016 008 IF(LNS .LT. 52) GO TO 18
000017 007 PRINT 2
000018 007 LNS = 0
000019 007 18 PRINT 3, (IN(1),15,11),ZIN
000020 007 3 FORMAT(2X,15,4X,13,3X,13,5X,11,3X,13,2X,13,2X,15,3(2X,F9.3))
000021 007 LNS = LNS + 2
000022 007 C
000023 007 50 1CT = 1CT + 1
000024 007 DO 55 1 = 1,4
000025 007 PRT(1,1CT) = IN(1)
000026 007 IF(1CT .NE. 4) GO TO 10
000027 007 IF(LNS .LT. 54) GO TO 60
000028 007 PRINT 2
000029 007 LNS = 0
000030 007 60 PRINT 5, PRT
000031 007 5 FORMAT(10X,4(5X,24(1X,A6,1X,15))
000032 007 1CT = 0
000033 007 LNS = LNS + 1
000034 007 GO TO 10
000035 007 70 IF(LNS .LT. 54) GO TO 75
000036 007 PRINT 2
000037 007 LNS = 0
000038 007 75 PRINT 5, ((PRT(I,J),I=1,4),J=1,1CT)
000039 007 1CT = 0
000040 007 LNS = LNS + 1
000041 007 GO TO 15
000042 011 80 IF(1CT .EQ. 0) GO TO 100
000043 011 IF(LNS .LT. 54) GO TO 85
000044 011 PRINT 2
000045 011 LNS = 0
000046 011 85 PRINT 5, ((PRT(I,J),I=1,4),J=1,1CT)
000047 011 LNS = LNS + 1
000048 007 C PROCESS NUR PACKAGES
000049 007 100 PRINT 101
000050 007 101 FORMAT(1H1,' PACKAGE DEST ORIG MODE RDD AVL ENTRY '
000051 007 * ,TOT PAX TOT STON TOT BARRL ,22H1000'S OF PAX, TONS,
000052 007 * , BARRELS,/)
000053 007 LNS = 0
000054 007 110 READ(19,11,END=1000) JN,ZJN
000055 007 111 FORMAT(4(2X,15),2X,13,3X,13,3X,15,3(2X,F9.3))

```

••UNCLASSIFIED••

MRG POST-PROCESSOR PROGRAM PRTPCK

```

000056 008 IF(LNS .LT. 52) GO TO 120
000057 007 LNS = 0
000058 007 PRINT 101
000059 008 120 LNS = LNS + 2
000060 007 PRINT 3,JN,ZJN
000061 007 GO TO 110
000062 007 1000 STOP
000063 007 END
    
```

END ELT.

••UNCLASSIFIED••

MRG POST-PROCESSOR PROGRAM BUILDMRG

••UNCLASSIFIED••

MRG POST-PROCESSOR PROGRAM PRTPCK

●●UNCLASSIFIED●●

MRG POST-PROCESSOR PROGRAM BUILDMRG

REL.T.C	24MONTAGNE-BUILD-FASMRG	ELT007	RL1870	08/16-07:44:00-(1,1)
000001	000	010010	IDENTIFICATION DIVISION.	
000002	000	010020	PROGRAM-ID. 'BUILD'.	
000003	000	010030	AUTHOR. J MEYEROWITZ.	
000004	000	010040	DATE WRITTEN. 02 MAY 1973.	
000005	000	010060	REMARKS. READ A FAS FILE FROM TAPE	
000006	000	010070		307 CHARACTERS BLOCKED 24 PER RECORD.
000007	000	010080	WRITE A UDF FILE TO MASS STORAGE USING ISFM	
000008	000	010090		360 CHARACTERS PER RECORD.
000009	000	020010	ENVIRONMENT DIVISION.	
000010	000	020020	CONFIGURATION SECTION.	
000011	000	020030	SOURCE-COMPUTER. UNIVAC-1108.	
000012	000	020040	OBJECT-COMPUTER. UNIVAC-1108.	
000013	000	020050	INPUT-OUTPUT SECTION.	
000014	000	020060	FILE-CONTROL.	
000015	000	020070	SELECT MASTER-FILE ASSIGN TO MASS-STORAGE FORCE	
000016	000	020080	ACCESS MODE IS RANDOM SYMBOLIC KEY IS REAL-KEY	
000017	000	020085	ORGANIZATION IS INDEXED FILE-DESCRIPTION IS FILE-DESCRIPT.	
000018	000	020120	SELECT FAST-FILE ASSIGN TO UNISERV FAS-FILE.	
000019	000	020130	SELECT PRINT-FILE ASSIGN TO PRINTER.	
000020	000	030010	DATA DIVISION.	
000021	000	030020	FILE SECTION.	
000022	000	030030	FD MASTER-FILE	
000023	000	030040	LABEL RECORDS ARE STANDARD	
000024	000	030050	DATA RECORD IS MASTER-REC.	
000025	000	030060	01 MASTER-REC	PICTURE IS X(10752).
000026	000	031010	FD FAS-FILE	
000027	000	031020	LABEL RECORD IS OMITTED	
000028	000	031030	RECORD CONTAINS 7488 CHARACTERS	
000029	000	031035	RECORDING MODE IS 1	
000030	000	031040	DATA RECORD IS TAPE-BLOCK.	
000031	000	031050	01 TAPE-BLOCK	PICTURE IS X(7488).
000032	000	032010	FD PRINT-FILE	
000033	000	032020	RECORD CONTAINS 132 CHARACTERS	
000034	000	032030	LABEL RECORD IS OMITTED	
000035	000	032040	DATA RECORD IS PRINT-LINE.	
000036	000	032050	01 PRINT-LINE	PICTURE IS X(132).
000037	000	040010	WORKING-STORAGE SECTION.	
000038	000	040020	77 NDEX	PICTURE IS 99 COMP SYNC RIGHT VALUE IS 0.
000039	000	040030	77 N	PICTURE IS H9 VALUE IS ZERO.
000040	000	040100	01 WORK-BLOCK.	
000041	000	040110	02 WORK-REC OCCURS 24 TIMES.	
000042	000	040120	03 PART-1.	
000043	000	040130	04 FILLER PICTURE IS X.	
000044	000	040140	04 WORKID.	
000045	000	040143	05 FICOD PICTURE IS X.	
000046	000	040144	05 COMPO PICTURE IS X.	
000047	000	040145	05 UICCC PICTURE IS X(16).	
000048	000	040146	05 EDATC PICTURE IS X(16).	
000049	000	040150	04 FILLER PICTURE IS X(11).	
000050	000	040160	03 PART-2.	
000051	000	04162	04 FILLER PIC X(12).	
000052	000		04 THEATER-CODE PIC X.	
000053	000		04 FILLER PIC X(13).	
000054	000		04 DSCMP PIC XX.	
000055	000		04 FILLER PIC XXX.	

```
000056 000 04164 04 ESCON PIC X*
000057 000 04166 04 FILLER PIC X(108).
000058 000 04 TRON PIC X*
000059 000 04 FILLER PIC X(127).
000060 000 050010 01 DATA-REC. PICTURE IS X(26).
000061 000 050020 02 SEG-1 PICTURE IS X
000062 000 050030 02 FILLER PICTURE IS X(333).
000063 000 050040 02 SEG-2 VALUE IS 'X'.
000064 000 060010 01 PRT-AREA. PICTURE IS X(16) OCCURS 7 TIMES.
000065 000 060020 02 PRT-ITEM
000066 000 080010 01 ER-LINE. PICTURE IS X(12) VALUE IS 'INVALID KEY '.
000067 000 080020 02 FILLER PICTURE IS X(14).
000068 000 080030 02 ER-KEY
000069 000 150010 01 REAL-KEY. PICTURE IS X(14).
000070 000 150020 02 REC-KEY PICTURE IS X(4).
000071 000 150030 02 FILLER
000072 000 180010 01 FILE-DESCRIPT. PICTURE IS H9(10) VALUE IS 12000.
000073 000 180020 02 NUMBER-OF-RECORDS PICTURE IS H9(10) VALUE IS 360.
000074 000 180030 02 RECORD-LENGTH PICTURE IS H9(10) VALUE IS 0.
000075 000 180040 02 RECORD-LENGTH-MAX PICTURE IS H9(10) VALUE IS 18.
000076 000 180050 02 RECORD-KEY-LENGTH PICTURE IS H9(10) VALUE IS 10.
000077 000 180060 02 OVERFLOW-AREA PICTURE IS H9(10) VALUE IS 1000.
000078 000 180070 02 1-O-F
000079 000 180080 02 DATA-NAME-8. PICTURE IS H9(10) VALUE IS 1000.
000080 000 180090 03 DATA-NAME-9 PICTURE IS X(3).
000081 000 180100 03 DATA-NAME-10 PICTURE IS X(3).
000082 000 190010 01 INFORM.
000083 000 190020 02 NUMBER-OF-BLOCKS PICTURE IS H9(10).
000084 000 190030 02 NUMBER-OF-INDX-BLOCKS PICTURE IS H9(10).
000085 000 190040 02 NUMBER-OF-OVERFL-BLOCKS PICTURE IS H9(10).
000086 000 190050 02 NUMBER-OF-RECORDS PICTURE IS H9(10).
000087 000 190060 02 NUMBER-OF-RECORDS-IN-10F PICTURE IS H9(10).
000088 000 190070 02 NUMBER-OF-RECORDS-DELT0 PICTURE IS H9(10).
000089 000 190080 02 NUMBER-OF-RECORDS-READ PICTURE IS H9(10).
000090 000 190090 02 NUMBER-OF-RECORDS-READ-FROM-IF PICTURE IS H9(10).
000091 000 190100 02 NUMBER-OF-RECORDS-WRITTEN PICTURE IS H9(10).
000092 000 195010 01 INFORM-FL-DATA.
000093 000 195020 02 NUMBER-OF-BLOCKS PICTURE IS 9(6).
000094 000 195030 02 NUMBER-OF-INDX-BLOCKS PICTURE IS 9(6).
000095 000 195040 02 NUMBER-OF-OVERFL-BLOCKS PICTURE IS 9(6).
000096 000 195050 02 NUMBER-OF-RECORDS PICTURE IS 9(6).
000097 000 195060 02 NUMBER-OF-RECORDS-IN-10F PICTURE IS 9(6).
000098 000 195070 02 NUMBER-OF-RECORDS-DELT0 PICTURE IS 9(6).
000099 000 195080 02 NUMBER-OF-RECORDS-READ PICTURE IS 9(6).
000100 000 195090 02 NUMBER-OF-RECORDS-READ-FROM-IF PICTURE IS 9(6).
000101 000 195100 02 NUMBER-OF-RECORDS-WRITTEN PICTURE IS 9(6).
000102 000 200000 PROCEDURE DIVISION.
000103 000 200010 START-TEST.
000104 000 200020 OPEN INPUT FAS-FILE WITH NO REWIND.
000105 000 200030 OPEN OUTPUT PRINT-FILE.
000106 000 200040 OPEN OUTPUT MASTER-FILE.
000107 000 210000 TAPE-READ.
000108 000 210010 MOVE ZERO TO NDEX.
000109 000 210020 READ FAS-FILE AT END GO TO FINI.
000110 000 210030 MOVE TAPE-BLOCK TO WORK-BLOCK.
000111 000 220000 GET-ITEM.
000112 000 220010 ADD 1 TO NDEX.
```

```
000113 000 220020 IF NDEX = 25 GO TO TAPE-READ.
000114 000 220021 IF WORKID (INDEX) = SPACES GO TO GET-ITEM.
000115 000 220025 IF WORKID (INDEX) = '9999999999' GO TO GET-ITEM.
000116 000 MOVE THEATER-CODE (INDEX) TO TRCON (INDEX).
000117 000 IF COMPO (INDEX) = '9' MOVE 'DC' TO USCMP (INDEX).
000118 000 MOVE PART-1 (INDEX) TO SEG-1.
000119 000 MOVE PART-2 (INDEX) TO SEG-2.
000120 000 MOVE WORKID (INDEX) TO REAL-KEY.
000121 000 ADD 1 TO N.
000122 000 IF N = 8 PERFORM LINE-LIST THRU X-LIST.
000123 000 WRITE MASTER-REC FROM DATA-REC INVALID KEY GO TO ERR-0.
000124 000 MOVE REAL-KEY TO PRT-ITEM (N).
000125 000 GO TO GET-ITEM.
000126 000 220080 LINE-LIST.
000127 000 225010 MOVE 1 TO N.
000128 000 225020 WRITE PRINT-LINE FROM PRT-AREA.
000129 000 225030 WRITE PRINT-LINE FROM PRT-AREA.
000130 000 225040 MOVE SPACES TO PRT-AREA.
000131 000 225050 X-LIST.
000132 000 225060 EXIT.
000133 000 226010 ERR-0.
000134 000 MOVE REAL-KEY TO ER-KEY.
000135 000 226025 WRITE PRINT-LINE FROM ER-LINE.
000136 000 226030 GO TO FINI.
000137 000 230000 FINI.
000138 000 230020 CLOSE MASTER-FILE USING INFORM.
000139 000 230030 MOVE CORRESPONDING INFORM TO INFORM-FL-DATA.
000140 000 230035 WRITE PRINT-LINE FROM PRT-AREA.
000141 000 230040 WRITE PRINT-LINE FROM INFORM-FL-DATA AFTER 2 LINES.
000142 000 230050 CLOSE PRINT-FILE.
000143 001 230010 CLOSE FAS-FILE.
000144 000 230060 STOP RUN.
```

END ELT.

DBKPT PRINTS

COMPUTER ASSISTED MATCH PROGRAM (CAMP)

CHAPTER VIII
SAMPLE OUTPUT

VIII-1

ORUN/PRNT 8124E,H3883P304JK,UNCLASSIFIED,100,300 • RSORT/TEST

ORHG RSORT/TEST *** UNCLASSIFIED ***

ORMSG,N INPUT FILES

ORQUAL UNCLASSIFIED

ORASG,A •24TESTRGR• • CAMPSUN, REQUIREMENTS FILE FROM FASTALS

ORUSE 14,•24TESTRGR•

ORASG,T 15

VIII-3

ORED 24TESTDATA,LOUIN,15• • UIN TO BE DELETED FROM REQUIREMENTS FILE
CASE UPPER ASSUMED
ED 15:00-08/19/76-10:23:40-(2.)
EDIT
LINES:2 FIELDATA

ORMSG,N OUTPUT FILES

ORDELETE,C 24UIN•
FURPUR 27R1 RL71-3 08/19/76 10:23:42

ORDELETE,C 24REQ•
24REQ 15 NOT CATALOGUED OR ASSIGNED
FAC STATUS: 400010000000

ORDELETE,C 24TDA•

ORASG,UP 24UIN,•F40,844UB • REQUIREMENTS DELETED BASED ON UIN

PRECEDING PAGE BLANK-NOT FILMED

*** UNCLASSIFIED ***

RSORT/TEST

0156,UP 24REQ,0F40,8440B • REQUIREMENTS TO BE MATCHED

0156,UP 24TDA,0F40,8440C • TDA UNITS DELETED FROM REQUIREMENTS

0USE 11,24UIN.

0USE 12,24REQ.

0USE 13,24TDA.

0156,A 0F5RS.

0USE 1PFS,0F5RS.

0XGT

*** UNCLASSIFIED ***

RSORT/TEST

FIRST TIME PERIOD = 1

... HAND-PLAYED-UIN-TABLE ...

SEQ UIN TP

1 4 0

SEQ	UIC	TP	COUNT
1	4	0	1

VIII-6

RSORT/TEST *** UNCLASSIFIED ***

*** TOTALS ***

HAND-PLAYED-UNIT = 1
TDA-UNITS = 1
REQUIREMENTS = 18
TOTAL = 20

SHDG RSORT UIN FILE *** UNCLASSIFIED ***

DATA: L 11.
DATA T7 RL70-5 08/19-10:24:12
1. 17052 00 5 H0 0 0 1 HMC ARMO CAV RG T W/AT 1000 2 2 00000 00030 00040 00188 00238 4 AR 238
END DATA.

SHDG RSORT REQ FILE *** UNCLASSIFIED ***

VIII-7
DATA: L 12.
DATA T7 RL70-5 08/19-10:24:12
1. 5035 99 2 H5 0 0 1 BN COMBAT 0010 2 1 2 31300 00040 00000 00745 00785 270 EN 785 100
2. 5035 99 2 H5 0 0 1 BN COMBAT 0010 2 1 2 31300 00040 00000 00745 00785 270 EN 785 200
3. 5035 99 3 H5 0 0 1 BN COMBAT 0010 2 1 2 31300 00040 00000 00745 00785 270 EN 785 300
4. 5115 99 2 H5 0 0 1 BN COMBAT H VY 0011 5 1 2 31300 00043 00000 00804 00847 290 EN 847 400
5. 5115 99 2 H5 0 0 1 BN COMBAT H VY 0011 5 1 2 31300 00043 00000 00804 00847 290 EN 847 500
6. 5115 99 3 H5 0 0 1 BN COMBAT H VY 0011 5 1 2 31300 00043 00000 00804 00847 290 EN 847 600
7. 5115 99 3 H5 0 0 1 BN COMBAT H VY 0011 5 1 2 31300 00043 00000 00804 00847 290 EN 847 700
8. 5115 99 4 H5 0 0 1 BN COMBAT H VY 0011 5 1 2 31300 00043 00000 00804 00847 290 EN 847 800
9. 5115 99 4 H5 0 0 1 BN COMBAT H VY 0011 5 1 2 31300 00043 00000 00804 00847 290 EN 847 900
10. 5327 99 4 G6 0 0 1 CO TOPOGR APHC CORPS 0010 2 1 2 31531 00004 00004 00138 00146 308 EN 147 1000
11. 5510 FA 2 H2 0 1 1 TM FIREF IGH TIN G HQ 0010 5 1 2 31572 00001 00000 00003 00004 324 EN 4 11FA
12. 5510 FA 2 H2 0 1 1 TM FIREF IGH TIN G HQ 0010 5 1 2 31572 00001 00000 00003 00004 324 EN 4 12FA
13. 5510 FB 2 H2 0 1 1 TM FIRE TRUCK 0010 3 1 2 31300 00000 00000 00006 00006 322 EN 6 13FB
14. 5510 FB 2 H2 0 1 1 TM FIRE TRUCK 0010 5 1 2 31300 00000 00000 00006 00006 324 EN 6 14FB
15. 5510 FB 2 H2 0 1 1 TM FIRE TRUCK 0010 5 1 2 31300 00000 00000 00006 00006 324 EN 6 15FB
16. 5510 FB 3 H2 0 1 1 TM FIRE TRUCK 0010 5 1 2 31300 00000 00000 00006 00006 324 EN 6 16FB
17. 5510 FB 3 H2 0 1 1 TM FIRE TRUCK 0010 5 1 2 31300 00000 00000 00006 00006 324 EN 6 17FB
18. 5510 FB 3 H2 0 1 1 TM FIRE TRUCK 0010 5 1 2 31300 00000 00000 00006 00006 324 EN 6 18FB
END DATA.

SHDG RSORT TDA FILE *** UNCLASSIFIED ***

DATA: L 13.
DATA T7 RL70-5 08/19-10:24:13
1. 0 1 1 HMC THEAT ER ARM Y 0100 5 2 2 39900 00045 00000 00839 00884 2 HJ 884
END DATA.

RSORT/TEST *** UNCLASSIFIED ***

WPT.1
 FURPUR 27R1 RL71-3 08/19/76 10:24:13
 UNCLASSIFIED*TPF*10).F40/30.T
 UNCLASSIFIED*24TESTRUR*1).F40/30.A 14.
 UNCLASSIFIED*1510).F40/30.T
 UNCLASSIFIED*ED*TC612ME10).F40/30.T ED*TC.
 UNCLASSIFIED*24UIN11).F40/30.C 11.
 UNCLASSIFIED*24REW11).F40/30.C 12.
 UNCLASSIFIED*24TDA11).F40/30.C 13.
 UNCLASSIFIED*F5RS11).F40/30.A TPF*,
 UNCLASSIFIED*XA10).F40/30.T SA,
 UNCLASSIFIED*XB10).F14/24.T SB.

WSTART START*24RTEST*FSORT/TEST

8-111A
 01111

GRUN/PRNT B124A.H3883P3043K1UNCLASSIFIED.100.500 • FSORT/TEST

MSG FSORT/TEST *** UNCLASSIFIED ***

MSG,N INPUT FILES

MSG,A •J2TESTFORCE• • FAS FILE OUTPUT BY SORTJDS

MSG 2. •J2TESTFORCE•

MSG,T 14,F40 • HAND PLAYED UNIT DATA

MSG,T 15,F40 • LOCKED OUT UNIT DATA

MSG,T 16,F40 • ALTERNATE THEATER UNIT DATA

MSG,T 17,F40 • M-CODE 1 (POMCUS) UNIT DATA

MSG,T 18,F40 • M-CODE 0 (IN-COUNTRY) UNIT DATA

MSG,T 19,F40 • LOCK OUT COMPO DATA

MSG TPFS,24TESTDATA.

MSG,N LOAD DATA INTO FILES

QED HPUIC,14.
CASE UPPER ASSUMED
ED 15.00-08/19776-10:24:42-(1.)
EDIT
LINES:1 FIELDATA

BED LORD1.15.
 CASE UPPER ASSUMED
 ED 15.00-08/19/76-10:24:46-(0.)
 EDIT
 LINES:1 FIELDATA

BED ALTHR.16.
 CASE UPPER ASSUMED
 ED 15.00-08/19/76-10:24:52-(0.)
 EDIT
 LINES:1 FIELDATA

BED RCMCI.17.
 CASE UPPER ASSUMED
 ED 15.00-08/19/76-10:24:55-(0.)
 EDIT
 LINES:1 FIELDATA

BED ICMCO.18.
 CASE UPPER ASSUMED
 ED 15.00-08/19/76-10:24:58-(0.)
 EDIT
 LINES:2 FIELDATA

VIII-10

BED LOCMP.19.
 CASE UPPER ASSUMED
 ED 15.00-08/19/76-10:25:01-(0.)
 EDIT
 LINES:2 FIELDATA

PMMSG.N OUTPUT FILES

BDELETE.C 24LOU.
 FURPUR 27RI RL71-3 08/19/76 10:25:03

BDELETE.C 24ATL.

BDELETE.C 24ALT.

0DELETE,C 24HPU.

0DELETE,C 24CAN.

0ASG,UP 24LOU.,F40,8440C • LOCKED OUT UNIT FILE

0ASG,UP 24ATL.,F40,8440B • ABOVE-THE-LINE UNIT FILE

0ASG,UP 24ALT.,F40,8440B • ALTERNATE THEATER UNIT FILE

0ASG,UP 24HPU.,F40,8440B • HAND PLAYED UNIT FILE

0ASG,UP 24CAN.,F40,8440C • MATCH CANDIDATE FILE

0USE 31,24LOU.

0USE 32,24ATL.

0USE 33,24ALT.

0USE 34,24HPU.

0USE 13,24CAN.

0FREE,A TPF3

0QUAL UNCLASSIFIED

0ASG,A •F5F5.

DATE 081976

FSORT/TEST *** UNCLASSIFIED ***

0USE TPFS, *FSFS.

0XQT

VIII-12

DATE 081976

FSORT/TEST *** UNCLASSIFIED ***

*** HAND-PLAYED-UNITS-TABLE ***

SEQ	UIC	ROBCO	ADCO1	ADCO2	ADCO3
1	TEST24		1		US9

*** LOCKOUT TABLE ***

SEQ ADC01

1 2

FSORT/TEST ... UNCLASSIFIED ...

... ALTERNATE-THEATER-TABLE ...

SEQ ADC01

1 4

FSORT/TEST *** UNCLASSIFIED ***

*** MCODE # 1 TABLE ***

SEQ ROBCO RDD/ATL RDD/BTL

1 ROB 075

FSORT/TEST *** UNCLASSIFIED ***

*** MCODE = 0 TABLE ***

SEQ	COMPO
1	000
2	7

FSORT/TEST *** UNCLASSIFIED ***

*** HAND-PLAYED-COMPOS-TABLE ***

SEW COMPO

1 4
2 5

*** TOTALS ***

LOCKOUT = 1
 ABOVE-THE-LINE = 5
 ALT-THEATER = 2
 HAND-PLAYED-UNITS = 2
 MATCH-CANDIDATES = 13
 TOTAL = 23

MDG,P L O C K O U T ... UNCLASSIFIED ...

31. @DATA,L

DATA T7 RL70-5 08/19-10:25:41

1. 55128002 T10040TESTUHS200TCCO MEDIUM BOAT 35091XXATACOMA 6HADD0174206130 1ADCO1
 END DATA.

MDG,P A B O V E - T H E - L I N E ... UNCLASSIFIED ...

32. @DATA,L

DATA T7 RL70-5 08/19-10:25:42

1. 170520013T10000TESTU5H0100ARHMT RGT CAVALRY 18003XXAFT BLISS 5TX00241102018 1NO NO
 2. 170551013T10010TESTU6H0100ARSW CAVALRY 18003XXAFT BLISS 5TX00936102018 2NO NO
 3. 170551013T10010TESTU7H0100ARSW CAVALRY 18003XXAFT BLISS 5TX00936102018 3NO NO
 4. 170551013T10010TESTU8H0100ARSW CAVALRY 18003XXAFT BLISS 5TX00936102018 4NO NO
 5. 173870013T10020TESTU9H5100ARCO ATK MEL 18003XXAFT H000 5TX00359100000 5NO NO
 END DATA.

MDG,P A L T E R N A T E - T H E A T E R ... UNCLASSIFIED ...

33. @DATA,L

DATA T7 RL70-5 08/19-10:25:43

1. 01127001 T10000TESTU3H1100AVCO CORPS 30023XXAFT BRAGG INCO0110400999 1
 2. 01207001 T10000TESTU4H2100AVU AIR TRAFFIC CON 30012XXAFT BRAGG INCO0339403999 2
 END DATA.

MDG,P H A N D - P L A Y E D ... UNCLASSIFIED ...

34. @DATA,L

DATA T7 RL70-5 08/19-10:25:43

1. 050790011T10040TEST20H4100ENCO ASLT FLT BRG RIB 21435R0BFT BLV0IR1VAD00176103075 1

2. US1240024T10080TEST24G680UENCO DUMP TRUCK J1436XXKOROFINO 610000J5104059 2

END DATA.

BHOG.P M A T C H - C A N D I D A T E *** UNCLASSIFIED ***

DATA.L 13.

DATA	17	RL70-5	08/19-10:25:43	TEST	1H5100ENBN	COMBAT	ARMY	21353XXXFT	L	WOODSMO080U8102015	100
1.	050359913310020	TEST1H5100ENBN	COMBAT	ARMY	21353XXXFT	CAMPBLIKY0080U8102021	200				
2.	050359913310030	TEST1H5100ENBN	COMBAT	ARMY	21353XXXFT	CAMPBLIKY0080U8102021	200				
3.	050359924310050	TEST1H5100ENBN	COMBAT	ARMY	21353XXXFT	65000785103036	300				
4.	050359924310050	TEST1H5100ENBN	COMBAT	ARMY	21353XXXFT	65000785103036	400				
5.	050359924310060	TEST1H5100ENBN	COMBAT	ARMY	21353XXXFT	65000785103036	500				
6.	051159913310030	TEST1H5100ENBN	COMBAT	ARMY	21364XXXFT	65000785104071	600				
7.	051159924310060	TEST1H5100ENBN	COMBAT	ARMY	21364XXXFT	65000785104071	700				
8.	051159924310070	TEST1H5100ENBN	COMBAT	ARMY	21364XXXFT	65000785104071	800				
9.	051159924310070	TEST1H5100ENBN	COMBAT	ARMY	21364XXXFT	65000785104071	900				
10.	051159924310080	TEST1H5100ENBN	COMBAT	ARMY	21364XXXFT	65000785104071	1000				
11.	053279934310090	TEST1H5100ENBN	COMBAT	ARMY	21364XXXFT	65000785104071	1100				
12.	053279934310100	TEST1H5100ENBN	COMBAT	ARMY	21364XXXFT	65000785104071	1200				
13.	05510F434310100	TEST1H5100ENBN	COMBAT	ARMY	21364XXXFT	65000785104071	13FA				

END DATA.

0PRT.1
FURPUR 27R1 RL71-J 08/19/76 10:25:44
UNCLASSIFIED*TPF8(U),F40/30,T
UNCLASSIFIED*32TESTFORCE(1),F40/30,A 2,
UNCLASSIFIED*14(U),F40/30,T
UNCLASSIFIED*15(U),F40/30,T
UNCLASSIFIED*16(U),F40/30,T
UNCLASSIFIED*17(U),F40/30,T
UNCLASSIFIED*18(U),F40/30,T
UNCLASSIFIED*19(U),F40/30,T
UNCLASSIFIED*ED8TCB124A(U),F40/30,T ED8TC,
UNCLASSIFIED*24L00(1),F40/30,C 31,
UNCLASSIFIED*24ATL(1),F40/30,C 32,
UNCLASSIFIED*24ALT(1),F40/30,C 33,
UNCLASSIFIED*24HPU(1),F40/30,C 34,
UNCLASSIFIED*24CAN(1),F40/30,C 13,
UNCLASSIFIED*F5F5(1),F40/30,A TPF5,
UNCLASSIFIED*XA(U),F40/30,T 8A,
UNCLASSIFIED*XB(U),F14/24,T 8B,

0START START*24RTEST*ATL/TEST

BRUN./PRNT B124B.H3883P6141,UNCLASSIFIED,30,100

SHDG ATL/TEST ... UNCLASSIFIED ...

MSG,N INPUT FILES

MSG,T 11

MSG,T 12

QED 24TESTDATA.SPRDD,11.
CASE UPPER ASSUMED
ED 15.00-08/19/76-11:25:07-(2,)
EDIT
LINES:10 FIELDATA

VIII-21

QED 24TESTDATA.ATLRDD,12.
CASE UPPER ASSUMED
ED 15.00-08/19/76-11:25:09-(1,)
EDIT
LINES:1 FIELDATA

MSG,A 24ATL. • ABOVE-THE-LINE FILE FROM FSORT

QUSE 13,24ATL.

MSG,N OUTPUT FILES

QDELETE,C 24MATL.
FURPUR 27R1 RL71-3 08/19/76 11:25:10

MSG,UP 24MATL.,F40.8440B • MODIFIED ABOVE-THE-LINE FILE (W/RDD)

*** UNCLASSIFIED ***

ATL/TEST
WUSE 14.24MATL.

WASG.A *FSATL.

WUSE 14.24MATL.

WAXQT
1 0 0 0 0 0 0 0 0 0
2 2 5 8 11 14 17 17
3 20 23 26 29 32 35 35
4 38 41 44 47 50 53 53
5 56 59 62 65 68 71 71
6 74 77 80 83 86 89 89
7 99 99 99 99 99 99 99
8 99 99 99 99 99 99 99
9 99 99 99 99 99 99 99
10 99 99 99 99 99 99 99
1 18003 1 059 05
*** ABOVE-THE-LINE FINI ***

VI:1-22

*** UNCLASSIFIED ***

WHDG ATL RESULTS

WHDG 14.
DATA 17 RL70-5 08/19-11:25:16
1. 17052001311000TEST05H0100ARHMT RGT CAVALRY 18003XXAFT BLISS 5TX00241105059 1 NO
2. 17055101311001TEST06H0100ARSW CAVALRY 18003XXAFT BLISS 5TX00936105059 2 NO
3. 17055101311001TEST07H0100ARSW CAVALRY 18003XXAFT BLISS 5TX00936105059 3 NO
4. 17055101311001TEST08H0100ARSW CAVALRY 18003XXAFT BLISS 5TX00936105059 4 NO
5. 17387001311002TEST09H5100ARCO ATK HEL 18003XXAFT H000 5TX00359105059 5 NO
END DATA.

WPR1, I
FURPUR 27R1 RL71-3 08/19/76 11:25:17
UNCLASSIFIED*TPF8(0),F40/30,T
UNCLASSIFIED*11(0),F40/30,T
UNCLASSIFIED*12(0),F40/30,T
UNCLASSIFIED*ED8TCB124B(0),F40/30,T ED8TC,
UNCLASSIFIED*24MATL(1),F40/30,A 13,
UNCLASSIFIED*24MATL(1),F40/30,C 14,
UNCLASSIFIED*FSATL(1),F40/30,A TPF8,

*** UNCLASSIFIED ***

ATL/TEST

BRUN, /PRNT B124C, H3883P6141, UNCLASSIFIED, 30, 100 *ALT/TEST

MSG ALT/TEST *** UNCLASSIFIED ***

MSG, N INPUT FILES

MSG, T 11

ED 24TESTDATA, ALTHR, 11, * ALTERNATE THEATER DATA
CASE UPPER ASSUMED
ED 15, 00-08/19/76-10:26:34-(0,1)
EDIT
LINES: 1 FIELD DATA

MSG, A 24ALT.

USE 12, 24ALT.

MSG, N OUTPUT FILES

DELETE, C 24MALT.
FURPUR 27R1 RL71-3 08/19/76 10:26:36

MSG, UP 24MALT., F40, 84408 * ALTERNATE THEATER FILE MODIFIED (W/RDD)

USE 13, 24MALT.

QUAL UNCLASSIFIED

MSG, A *F5ALT.

*** UNCLASSIFIED ***

ALT/TEST

WUSE TPFS, OFSALT.

0XGT

VIII-24

*** UNCLASSIFIED ***

ALT/TEST

ALT/TEST *** UNCLASSIFIED ***

ADCO1 LOWER-LIMIT UPPER-LIMIT INCRE NEW-ADCO1

4 010 020 002 4

*** ALTERNATE-THEATER FINI ***

SHDG ALT RESULTS *** UNCLASSIFIED ***

DATA, L 13.

DATA T7 RL70-5 08/19-10:26:40

1. 01127001 T10000TEST03H1100AVCO CORPS

2. 01207001 T10000TEST04H2100AVU AIR TRAFFIC CON

30623XXXFT BRAGG INCO0110400010 1
30612XXXFT BRAGG INCO0339400012 2

END DATA.

GPRT.1
FURPUR 27R1 RL71-3 08/19/76 10:26:40
UNCLASSIFIED*TPFS(0),F40/30,T
UNCLASSIFIED*11(0),F40/30,T
UNCLASSIFIED*ED8TCB124C(0),F40/30,T ED8TC,
UNCLASSIFIED*24ALT(1),F40/30,A 12,
UNCLASSIFIED*24HALT(1),F40/30,C 13,
UNCLASSIFIED*F5ALT(1),F40/30,A TPF\$,

VIII-25

START START*24HTEST*MATCH/TEST

GRUN 8124F,H3683P5141,UNCLASSIFIED,100,500 • MATCH/TEST

GRDG MATCH/TEST ••• UNCLASSIFIED •••

MSGIN INPUT FILES

BASG.A 24REQ. • REQUIREMENTS FILE FROM RSORT

BASG.A 24CAN. • MATCH CANDIDATES FILE FROM FSORT

BUSE 1,24CAN.

BUSE 2,24REQ.

BASG.T 11,F40 • PECOD DATA

BASG.T 12,F40 • SUPERIOD RUD DATA

BASG.T 13,F40 • NOTIONAL UNIT DATA

BUSE TPF8,24TESTDATA.

WED PECOD,11.
CASE UPPER ASSUMED
ED 15•00-08/19/76-10:27:09-(0,)
EDIT
LINES:23 FIELDATA

WED SPRUD,12.
CASE UPPER ASSUMED
ED 15•00-08/19/76-10:27:10-(1,)
EDIT
LINES:10 FIELDATA

WED NOTIN:13.
CASE UPPER ASSUMED
ED 15:00-08/19/76-10:27:12-(0.)
EDIT
LINES:20 FIELD:DATA

WFREE:1A TPF\$

WMSG:N OUTPUT FILES

WDELETE,C 24MOVR.
FURPUR 27RI RL71-3 08/19/76 10:27:13

WDELETE,C 24MREQ.

WDELETE,C 24MATCHRPT.

VIII-27

WASG:UP 24MOVR.,F40,8440A • THIS FILE NOT USED

WASG:UP 24MREQ.,F14,PACK61 • MATCH REQUIREMENTS (MATCHES AND NOTIONALS)

WASG:UP 24MATCHRPT.,F40 • MATCH REPORT FILE

WUSE 3,24MOVR.

WUSE 8,24MREQ.

WUSE 19,24MATCHRPT.

WQUAL UNCLASSIFIED

MATCH/TEST *** UNCLASSIFIED ***

0ASG.A *FSFM.

0USE TPFS.*FSFM.

0DELETE,C FSTPRT.
FURPUR 27R1 RL71-J 08/19/76 10:20:35

0ASG.UP FSTPRT.,F40 • INTERMEDIATE PRINT FILE

0BRKPT PRINTS/FSTPRT

VIII-28

REQ-SRC	FORCE-SRC	BRNCH	LEVEL	UNTOS	COMPU UIC	TPSN	UIN	TP	RSEQ	MC	ROBCO	STNM	LOCCO	RSTR	FSTR	FSEQ	
05035 00	05035 00	EN	BN	COMBAT ARMY	1	TEST10	21353	270	2	1	3	XXX	FT L WOOD	5MO	00785	00808	1
05035 00	05035 00	EN	BN	COMBAT ARMY	1	TEST11	21353	270	2	2	3	XXX	FT CAMPBL	1KY	00785	00808	2
05035 00	05035 00	EN	BN	COMBAT ARMY	2	TEST13	21353	270	3	3	4	XXX	HUKON	6SD	00785	00785	3
	05035 00	EN	BN	COMBAT ARMY	2	TEST12	21353					XXX	ASHLAND	5KY		00785	4
	05035 00	EN	BN	COMBAT ARMY	2	TEST14	21353					XXX	AMLCN FK	6UT		00785	5
05115 00	05115 00	EN	BN	COMBAT CHVYJ	1	TEST15	21364	290	2	4	3	XXX	FT RILEY	6KS	00847	00845	6
05115 00	05115 00	EN	BN	COMBAT CHVYJ	2	TEST16	21364	290	2	5	4	XXX	GULFPORT	1MS	00847	00847	7
05115 00	05115 00	EN	BN	COMBAT CHVYJ	2	TEST17	21364	290	3	6	4	XXX	PORTLAND	1ME	00847	00847	8
05115 00	05115 00	EN	BN	COMBAT CHVYJ	2	TEST18	21364	290	3	7	4	XXX	BINGHAMTN	1NY	00847	00847	9
05115 00	05115 00	EN	BN	COMBAT CHVYJ	2	TEST19	21364	290	4	8	4	XXX	BAI ROUGE	5LA	00847	00847	10
05115 00	05115 00	EN	BN	CMBT HVY	4	X50001	31300	290	4	9		XXX	FT RILEY	6KS	00847	00847	
05327 00	05327 00	EN	CO	TOPO CORPS	3	TEST21	31531	308	4	10	4	XXX	FT BLVDIR	1VA	00147	00147	11
	05327 00	EN	CO	TOPO CORPS	3	TEST22	31531					XXX	BELL	6CA		00147	12
05510 FA	05510 FA	EN	PLT	FIREFIGHTING	3	TEST23	31557	324	2	11	4	XXX	DANVERS	1MA	00004	00028	13
05510 FA	05510 FA	EN	TM	FIREFIGHTING HQ	4	X50002	31572	324	2	12		XXX	FT RILEY	6KS	00004	00004	
05510 FB	05115 00	EN	TM	FIRE TRUCK	4	X50003	31300	322	2	13			FT RILEY	6KS	00006	00006	
05510 FB	05115 00	EN	TM	FIRE TRUCK	4	X50004	31300	324	2	14			FT RILEY	6KS	00006	00006	
05510 FB	05115 00	EN	TM	FIRE TRUCK	4	X50005	31300	324	2	15			FT RILEY	6KS	00006	00006	
05510 FB	05115 00	EN	TM	FIRE TRUCK	4	X50006	31300	324	3	16			FT RILEY	6KS	00006	00006	
05510 FB	05115 00	EN	TM	FIRE TRUCK	4	X50007	31300	324	3	17			FT RILEY	6KS	00006	00006	
05510 FB	05115 00	EN	TM	FIRE TRUCK	4	X50008	31300	324	3	18			FT RILEY	6KS	00006	00006	

BRUN /PRNT 8124G.H3883P6141.UNCLASSIFIED.100.300 • LAYIN/TEST

0HDG LAYIN ••• UNCLASSIFIED •••

0MSGIN INPUT FILES

0ASG.T 10.

0ED 24TESTDATA.SPRDD.10.
CASE UPPER ASSUMED
ED 15.00-08/19/76-11:26:38-(2.)
EDIT
LINES:10 FIELDATA

0ASG.A 24MALT. • ALTERNATE THEATER UNITS FROM ALT

0ASG.A 24MATL. • ABOVE-THE-LINE UNITS FROM ATL

0ASG.A 24LOU. • LOCK OUT UNITS FROM FSORT

0ASG.A 24HPU. • HAND PLAYED UNITS FROM FSORT

0ASG.A 24MREQ. • MATCHES AND NOTIONALS FROM MATCH

0ASG.A 32TESTFORCE. • FORCE FILE FROM SORTUDS

0USE 11.24MALT.

0USE 12.24MATL.

0USE 13.24LOU.

*** UNCLASSIFIED ***

LAYIN

0USE 14.24HPU.

0USE 15.24HREG.

0USE 16.32TESTFORCE.

0MSG,N OUTPUT FILES

0DELETE,C 24FOUT.
FURPUR 27R1 RL71-3 08/19/76 11:28:12

0DELETE,C 24FPRT.

0ASG,UP 24FOUT.,F14,PACK62 • FORCE OVERLAID WITH MATCH RESULTS

0MSG,N 24FOUT WILL BE INPUT TO MRG

0ASG,UP 24FPRT.,F40 • INTERMEDIATE PRINT FILE

0USE 17.24FOUT.

0USE 18.24FPRT.

0GUAL UNCLASSIFIED

0ASG,A • F5LAY

0USE TPFS,•F5LAY

VIII-31

*** UNCLASSIFIED ***

LAYIN

0XGT 1 0 0 0 0 0 0 0 0 0
 2 2 5 8 11 14 17 17
 3 20 23 26 29 32 35 35
 4 38 41 44 47 50 53 53
 5 56 45 62 65 68 71 71
 6 74 77 80 83 86 89 89
 7 999 999 999 999 999 999 999
 8 999 999 999 999 999 999 999
 9 999 999 999 999 999 999 999
 10 999 999 999 999 999 999 999
 TALT= 2 TAILOV= 0 TATL= 5 TBYPAS= 0 TLKOUT= 1

THP= 2 TSHORT= 8 TMATCH= 10 TOVER= 3 TOTAL = 31

*** LAYIN FINI ***

0FREE 24FPRT.

0MSG.N 0SYM 24FPRT.

VIII-32

SEND
SEND IGNORED - IN CONTROL MODE

0PRT.1
 FURPUR 27R1 RL71-3 08/19/76 11:31:25
 UNCLASSIFIED*TPF8(0),F40/30,T
 UNCLASSIFIED*10(0),F40/30,T
 UNCLASSIFIED*ED8TC8124G(0),F40/30,T ED8TC,
 UNCLASSIFIED*24MALT(1),F40/30,A 11,
 UNCLASSIFIED*24MATL(1),F40/30,A 12,
 UNCLASSIFIED*24LOU(1),F40/30,A 13,
 UNCLASSIFIED*24HPU(1),F40/30,A 14,
 UNCLASSIFIED*24MREG(1),F14/24,A 15,
 UNCLASSIFIED*32TESTFORCE(1),F40/30,A 16,
 UNCLASSIFIED*24FOUT(1),F14/24,C 17,
 UNCLASSIFIED*F5LAY(1),F40/30,A TPF5,
 UNCLASSIFIED*XA(0),F40/30,T 8A,
 UNCLASSIFIED*XB(0),F14/24,T 8B,

0MSG.N START*24R*BINCOPY-SUM

BRUN, /R C124A, H38R3P5121, UNCLASSIFIED, 15,500 . GEORUN

BMHG GEO RUN

BASG, A 24FOUT. . FORCE FILE OUTPUT BY FMA PROGRAM LAYIN

BUSE 17, 24FOUT.

BQUAL UNCLASSIFIED

BASG, A *32GEO. . GEO FILE OUTPUT BY SORTARLOC

BASG, A *32NODE. . NODE TABLE

BASG, T 32F7. . PRINT OF MDI FILE

BASG, A 32F29 . MDI FILE - OUTPUT (SRC SEQUENCE)

BUSE 15, *32GEO.

BUSE 16, *32NODE.

BUSE 7, 32F7.

BUSE 29, 32F29.

BASG, XA *58CAMP.

BPRT, S *24TESTDATA, GFODATA

GEO RUN

FURPUR 27R1 RL71-3 08/25/76 08:58:24

DATE 082576

PAGE

3

VIII-34

GEO RUN

DATE 082576

PAGE

3

GEO RUN

UNCLASSIFIED*24TESTDATA(1),GEODATA

1 2
2 1 000 XPK 004 000
3 4 010 WTGR 012 010
4 1
5 18003 A HCRL

BIQT *58CAMP.GFOARS

SADD.P *24TESTDATA.GEODATA
NUMBER OF THEATERS 2
NUMBER OF EXCEPTIONS 1

YTHR AVAIL DEST(G) DEST(N) AVL(T)
1 0 XPK 4 0
4 10 WTGR 12 10

TPSN DEST(G) DFST(N) AVL R0D MODE THR ORIG(G) ORIG(N)
18003 0 0 A 0 HCRL 0

8800 RECORDS ON FILE BIGEO

83 RECORDS ON FILE BINODE
UNITS ACTIVE NO FIND RESERVE NO FIND AVE ACC/U
27 11 0 16 0 15.852
0

VIII-35

SPRT.1
FURPUR 27R1 RL71-1 08/25/76 09:00:23
UNCLASSIFIED*TPFS(0),F40/30,T
UNCLASSIFIED*24FOUT(1),F14/24,A 17,
UNCLASSIFIED*32GEO(1),F14/24,A 15,
UNCLASSIFIED*32MOOF(1),F14/24,A 16,
UNCLASSIFIED*32F7(0),F40/30,T 7,
UNCLASSIFIED*32F29(1),F40/30,A 29,
UNCLASSIFIED*58CAMP(1),F14/24,AX CSINTNAMES,
UNCLASSIFIED*XA(0),F40/30,T SA,
UNCLASSIFIED*XR(0),F14/24,T SB,
UNCLASSIFIED*21(0),F40/30,T

GED.R 32F7.
READ-ONLY MODE
CASE UPPER ASSUMED
EO 15.00-08/25/76-09:00:24-(0.)
EDIT

01127H10000 TEST03 30423 HCTL37 WTGR 190 12 10 10 P 4 3 CORPS
01207H20000 TEST04 30412 HCTL37 WTGR 190 12 10 P 4 3 IR TRAFFIC C
17052H00000 TEST05 18003 HCRL XPK 210 4 59 0 A 1 3 RGT CAVALRY
17055H01000 TEST06 18003 HCRL XPK 210 4 59 0 A 1 3 CAVALRY
17055H01000 TEST07 18003 HCRL XPK 210 4 59 0 A 1 3 CAVALRY
17055H01000 TEST08 18003 HCRL XPK 210 4 59 0 A 1 3 CAVALRY

GEO RUN

173R7H50000	TEST09 18003	HCRL	XPQK	210	4	59	0 A 1 3	ATK HEL	0
05035H50000	TEST10 21353	HGSH29	XPQK	186	4	2	0 P 1 3	COMBAT ARMY	0
05035H50000	TEST11 21353	H08L21	XPQK	227	4	2	0 P 1 3	COMBAT ARMY	0
05115H30000	TEST15 21364	HKB20	XPQK	178	4	5	0 P 1 3	COMBAT CHVY	0
05079H40000	TEST20 21435	HCHLS1	XPQK	192	4	75	0 A 1 1	ASLT FLT BRG	0
05035H50000	TEST13 21353	HFTZ48	XPQK	210	4	20	0 P 1 3	COMBAT ARMY	0
05115H30000	TEST16 21364	HFTZ48	XPQK	210	4	5	0 P 1 3	COMBAT CHVY	0
05115H30000	TEST17 21364	HFTZ48	XPQK	210	4	23	0 P 1 3	COMBAT CHVY	0
05115H30000	TEST18 21364	HFTZ48	XPQK	210	4	26	0 P 1 3	COMBAT CHVY	0
05115H30000	TEST19 21364	HFTZ48	XPQK	210	4	38	0 P 1 3	COMBAT CHVY	0
05124G60000	TEST24 31436	HFTZ48	XPQK	210	4	59	0 P 1 3	DUMP TRUCK	0
05327G60000	TEST21 31531	HFTZ48	XPQK	210	4	41	0 P 1 3	TOPO CORPS	0
05510H2FA00	TEST23 31557	HFTZ48	XPQK	210	4	8	0 P 1 3	FIREFIGHTIN	0
05115H50000	X50001 31300	HKB20	XPQK	178	4	44	0 P 1 3	CMBT HVY	0
05510H2FA01	X50002 31572	HKB20	XPQK	178	4	8	0 P 1 3	FIREFIGHTIN	0
05510H2F801	X50003 31300	HKB20	XPQK	178	4	11	0 P 1 3	FIRE TRUCK	0
05510H2F801	X50004 31300	HKB20	XPQK	178	4	14	0 P 1 3	FIRE TRUCK	0
05510H2F801	X50005 31300	HKB20	XPQK	178	4	17	0 P 1 3	FIRE TRUCK	0
05510H2F801	X50006 31300	HKB20	XPQK	178	4	29	0 P 1 3	FIRE TRUCK	0
05510H2F801	X50007 31300	HKB20	XPQK	178	4	32	0 P 1 3	FIRE TRUCK	0
05510H2F801	X50008 31300	HKB20	XPQK	178	4	35	0 P 1 3	FIRE TRUCK	0

EOF:27 SCAN:26

NO CORRECTIONS APPLIED.

0FREE 32F7.

0FREE 32F29.

0MSG.N START:24RTFST.LOGSUBRUN

QRUN./PRNT A124R.H3883P6141.UNCLASSIFIED:30,30 . LOGSUBRUN

QNDG UPDATING MDFI SRC'S - LOG SUB RUN

QQUAL UNCLASSIFIED

QASG.AX *32CAMP.

QASG.A 32F29. . MDFI FILE WITH BAD MATCH ON TUCHA

QUSE 29.,32F29.

QASG.A 32F29UP. . NEW MDFI WITH FIXED UP SRC'S

QUSE 25.,32F29UP.

QPRTS *24TESTDATA.LOGSUBDATA
FURPUR 27R1 RL71-3 08/24/76 18:59:19

UPDATING MDF1 SRC'S - LOG SUB RUN

UNCLASSIFIED*24TESTDATA(1).LOGSUBDATA

1
2 05115H450000 05115H30000

0X0T *32CAMP.UPMDF1

0ADD.P *24TESTDATA.LOGSUBDATA

0PRT.1
FURPUR 27R1 RL71-1 08/24/76 18:59:25
UNCLASSIFIED*TPFS(0).F40/30,T
UNCLASSIFIED*32CAMP(1).F14/24,AX CSINTNAMES,
UNCLASSIFIED*32F29(1).F40/30,A 29,
UNCLASSIFIED*32F29UP(1).F40/30,A 25,
UNCLASSIFIED*XA(0).F40/30,T SA,
UNCLASSIFIED*XR(0).F14/24,T SB,

0START START*24RTFST.LOGUPRUN

UPDATING MDF1 SRC'S - LOG SUB RUN

BRUN./PRNT A124C.H1883P6141,UNCLASSIFIED.50,50 . LOGUPRUN

SHDG LOG RUN

QUAL UNCLASSIFIED

QASG.XA .58CAMP.

QASG.A 32F29UP. . UPDATED MOF1 FILE

QASG.A .32TUSUM. . TUCHA FILE (SRC SEQUENCEF)

QASG.A 24POMSUM.

QASG.A 32F26. . MOF2 FILE - OUTPUT

QASG.A 32F11. . INTERMEDIATE FILE FOR RPG

QASG.A 32F10. . UNIT MORSA CARDS - OUTPUT

QUSE 11..32F11.

QUSE 10..32F10.

QUSE 29..32F29UP.

QUSE 28..32TUSUM.

QUSE 27..24POMSUM.

LOG RUN

USE 26.32F26.

08RT.S 024TESTDATA.LOGDATA
FURPUR 27R1 RL71-3 08/24/76 18:59:48

UNCLASSIFIED*2*TESTDATA*LOGDATA
 1 2 2 14 12
 2 200.0 100.0 25.0 25.0
 3 WADD,P 2*TESTDATA*LOGSUBDATA

WADD,P 2*TESTDATA*LOGSUBDATA

WADD,P 2*TESTDATA*LOGSUBDATA

NUMBER OF THEATERS 2
 NUMBER OF SETS 2
 TMTS SET PLANID

1 1
 4 2 2

AIRPORT	SEAPORT	NON-POMCUS ASR		POMCUS ASR	
		SUPPLY	AMMO	SUPPLY	AMMO
		200.000	25.000	.000	25.000

SRC LIST

05115H50000 05115H30000
 UNITS IN THEATER SETS 24 2 0 0 0 0
 FINAL SEQUENCE NUMBERS 24 2 0 0 0 0
 NO MATCHES 0
 UNITS IN THEATER SETS 25 2 0 0 0 0
 FINAL SEQUENCE NUMBERS 26 2 0 0 0 0
 NO MATCHES 0

VIII-41

SPRT,1
 FURPUR 27R1 RL71-3 08/24/76 19:00:25
 UNCLASSIFIED*TPFS(1),F40/30,T
 UNCLASSIFIED*SBRCAMP(1),F14/24,AX CSINTNAMES,
 UNCLASSIFIED*32F20UP(1),F40/30,A 29,
 UNCLASSIFIED*32TUSUM(1),F40/30,A 28,
 UNCLASSIFIED*24POMSUM(1),F40/30,A 27,
 UNCLASSIFIED*32F24(1),F40/30,A 26,
 UNCLASSIFIED*32F1(1),F40/30,A 11,
 UNCLASSIFIED*32F10(1),F40/30,A 10,
 UNCLASSIFIED*XA(1),F40/30,T 5A,

RED,P 10.
 READ-ONLY MODE
 CASE UPPER ASSUMED.
 ED 15.00-08/24/76-19:00:25-(0,)

EDIT
 0001A2A U7CORPS CO P 010010 HCTLWTGRWTGR0000102030623
 0001A2W0000100000210001080000028000000000000101127H10000041903AAE7TEST03
 0001A2FN 00016002839
 0002A2A U7IR TRAFFIC CU P 010012 HCTLWTGRWTGR0000338030612
 0002A2W000034000047000316000003000000000000401207H20000041903W9ACTEST04
 0002A2FN 000009001629

GRUN./PRNT A1240.H3883P6141.UNCLASSIFIED.100.300 * NURRUN

@HDG NUR RUN

@QUAL UNCLASSIFIED

@ASG.A 32F26. . MDF2 FILE (NON-POMCUS IN SRC / POMCUS IN UIC SEQ

@USE 26.32F26

@ASG.A 32F24. . NUR MORSA CARDS - OUTPUT (RDD SEQUENCE)

@USE 24.32F24

@ASG.A 32F23 . NUR DATA FOR PACKAGE - OUTPUT (RDD SEQUENCE)

@USE 23.32F23

@ASG.A 32F12. . INTERMEDIATE FILE FOR RPG

@USE 12.32F12

@ASG.XA *58CAMP

@PRT.S *24TESTDATA.NURDATA
FURPUR 27R1 RL71-3 08/24/76 19:00:46

NUR RUN

UNCLASSIFIED//24TESTDATA(1).NURDATA 12 2 2 14 12

[illegible]

VIII-45

9XQ7 • 58CAMP • NURARS

ADD,P	24	TESTDATA,NURDATA
NUMBER OF THEATERS	2	
NUMBER OF SETS	2	

7M7P SET

1

2

SET	PLAN	CDAY	RDAY	BLDUP	CDAY	ADAY	1	2	3	4	5	6	7	8	BLDUP	LEVELS	SAFE	LEVELS	
1	1	1	18	28	75	0	35	17	35	53	71	89	89	89	25	30	25	10	10
2	2	2	18	28	75	0	35	17	35	53	71	89	89	89	25	30	25	2	2

	SET	THEATER BASE	FILLER	FILLER	DAY
67	89	10	10	10	10

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	256	257	258	259	260	261	262	263	264	265	266	267	268	269	270	271	272	273	274	275	276	277	278	279	280	281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300	301	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319	320	321	322	323	324	325	326	327	328	329	330	331	332	333	334	335	336	337	338	339	340	341	342	343	344	345	346	347	348	349	350	351	352	353	354	355	356	357	358	359	360	361	362	363	364	365	366	367	368	369	370	371	372	373	374	375	376	377	378	379	380	381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399	400	401	402	403	404	405	406	407	408	409	410	411	412	413	414	415	416	417	418	419	420	421	422	423	424	425	426	427	428	429	430	431	432	433	434	435	436	437	438	439	440	441	442	443	444	445	446	447	448	449	450	451	452	453	454	455	456	457	458	459	460	461	462	463	464	465	466	467	468	469	470	471	472	473	474	475	476	477	478	479	480	481	482	483	484	485	486	487	488	489	490	491	492	493	494	495	496	497	498	499	500	501	502	503	504	505	506	507	508	509	510	511	512	513	514	515	516	517	518	519	520	521	522	523	524	525	526	527	528	529	530	531	532	533	534	535	536	537	538	539	540	541	542	543	544	545	546	547	548	549	550	551	552	553	554	555	556	557	558	559	560	561	562	563	564	565	566	567	568	569	570	571	572	573	574	575	576	577	578	579	580	581	582	583	584	585	586	587	588	589	590	591	592	593	594	595	596	597	598	599	600	601	602	603	604	605	606	607	608	609	610	611	612	613	614	615	616	617	618	619	620	621	622	623	624	625	626	627	628	629	630	631	632	633	634	635	636	637	638	639	640	641	642	643	644	645	646	647	648	649	650	651	652	653	654	655	656	657	658	659	660	661	662	663	664	665	666	667	668	669	670	671	672	673	674	675	676	677	678	679	680	681	682	683	684	685	686	687	688	689	690	691	692	693	694	695	696	697	698	699	700	701	702	703	704	705	706	707	708	709	710	711	712	713	714	715	716	717	718	719	720	721	722	723	724	725	726	727	728	729	730	731	732	733	734	735	736	737	738	739	740	741	742	743	744	745	746	747	748	749	750	751	752	753	754	755	756	757	758	759	760	761	762	763	764	765	766	767	768	769	770	771	772	773	774	775	776	777	778	779	780	781	782	783	784	785	786	787	788	789	790	791	792	793	794	795	796	797	798	799	800	801	802	803	804	805	806	807	808	809	810	811	812	813	814	815	816	817	818	819	820	821	822	823	824	825	826	827	828	829	830	831	832	833	834	835	836	837	838	839	840	841	842	843	844	845	846	847	848	849	850	851	852	853	854	855	856	857	858	859	860	861	862	863	864	865	866	867	868	869	870	871	872	873	874	875	876	877	878	879	880	881	882	883	884	885	886	887	888	889	890	891	892	893	894	895	896	897	898	899	900	901	902	903	904	905	906	907	908	909	910	911	912	913	914	915	916	917	918	919	920	921	922	923	924	925	926	927	928	929	930	931	932	933	934	935	936	937	938	939	940	941	942	943	944	945	946	947	948	949	950	951	952	953	954	955	956	957	958	959	960	961	962	963	964	965	966	967	968	969	970	971	972	973	974	975	976	977	978	979	980	981	982	983	984	985	986	987	988	989	990	991	992	993	994	995	996	997	998	999	1000	1001	1002	1003	1004	1005	1006	1007	1008	1009	1010	1011	1012	1013	1014	1015	1016	1017	1018	1019	1020	1021	1022	1023	1024	1025	1026	1027	1028	1029	1030	1031	1032	1033	1034	1035	1036	1037	1038	1039	1040	1041	1042	1043	1044	1045	1046	1047	1048	1049	1050	1051	1052	1053	1054	1055	1056	1057	1058	1059	1060	1061	1062	1063	1064	1065	1066	1067	1068	1069	1070	1071	1072	1073	1074	1075	1076	1077	1078	1079	1080	1081	1082	1083	1084	1085	1086	1087	1088	1089	1090	1091	1092	1093	1094	1095	1096	1097	1098	1099	1100	1101	1102	1103	1104	1105	1106	1107	1108	1109	1110	1111	1112	1113	1114	1115	1116	1117	1118	1119	1120	1121	1122	1123	1124	1125	1126	1127	1128	1129	1130	1131	1132	1133	1134	1135	1136	1137	1138	1139	1140	1141	1142	1143	1144	1145	1146	1147	1148	1149	1150	1151	1152	1153	1154	1155	1156	1157	1158	1159	1160	1161	1162	1163	1164	1165	1166	1167	1168	1169	1170	1171	1172	1173	1174	1175	1176	1177	1178	1179	1180	1181	1182	1183	1184	1185	1186	1187	1188	1189	1190	1191	1192	1193	1194	1195	1196	1197	1198	1199	1200	1201	1202	1203	1204	1205	1206	1207	1208	1209	1210	1211	1212	1213	1214	1215	1216	1217	1218	1219	1220	1221	1222	1223	1224	1225	1226	1227	1228	1229	1230	1231	1232	1233	1234	1235	1236	1237	1238	1239	1240	1241	1242	1243	1244	1245	1246	1247	1248	1249	1250	1251	1252	1253	1254	1255	1256	1257	1258	1259	1260	1261	1262	1263	1264	1265	1266	1267	1268	1269	1270	1271	1272	1273	1274	1275	1276	1277	1278	1279	1280	1281	1282	1283	1284	1285	1286	1287	1288	1289	1290	1291	1292	1293	1294	1295	1296	1297	1298	1299	1300	1301	1302	1303	1304	1305	1306	1307	1308	1309	1310	1311	1312	1313	1314	1315	1316	1317	1318	1319	1320	1321	1322	1323	1324	1325	1326	1327	1328	1329	1330	1331	1332	1333	1334	1335	1336	1337	1338	1339	1340	1341	1342	1343	1344	1345	1346	1347	1348	1349	1350	1351	1352	1353	1354	1355	1356	1357	1358	1359	1360	1361	1362	1363	1364	1365	1366	1367	1368	1369	1370	1371	1372	1373	1374	1375	1376	1377	1378	1379	1380	1381	1382	1383	1384	1385	1386	1387	1388	1389	1390	1391	1392	1393	1394	1395	1396	1397	1398	1399	1400	1401	1402	1403	1404	1405	1406	1407	1408	1409	1410	1411	1412	1413	1414	1415	1416	1417	1418	1419	1420	1421	1422	1423	1424	1425	1426	1427	1428	1429	1430	1431	1432	1433	1434	1435	1436	1437	1438	1439	1440	1441	1442	1443	1444	1445	1446	1447	1448	1449	1450	1451	1452	1453	1454	1455	1456	1457	1458	1459	1460	1461	1462	1463	1464	1465	1466	1467	1468	1469	1470	1471	1472	1473	1474	1475	1476	1477	1478</
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	--------

NUR RUN

DAY	THR	UNITS	STRENGTH	REPLACEMENTS	TYPE	DAYS ON	DAYS DESD	TONS ON	TONS REORD
1	1	0	0.						
1	2	0	0.						
18	1	9	3280.	0.	2	4.	10.	677.	963.
19	1	9	3280.	3.	2	9.	10.	1476.	164.
20	1	10	4060.	5.	2	7.	10.	1447.	583.
20	2	2	440.	1.	3	2.	2.	24.	0.
21	1	10	4060.	8.	2	9.	10.	1827.	203.
21	2	2	440.	1.	3	1.	2.	12.	0.
22	1	10	4060.	12.	2	9.	10.	1827.	203.
22	2	2	440.	1.	3	1.	2.	12.	0.
23	1	11	4907.	15.	2	7.	10.	1795.	658.
23	2	2	440.	2.	3	1.	2.	12.	0.
24	1	11	4907.	19.	2	9.	10.	2208.	245.
24	2	2	440.	2.	3	1.	2.	12.	0.
25	1	11	4907.	23.	2	9.	10.	2208.	245.
25	2	2	440.	2.	3	1.	2.	12.	0.
26	1	12	5754.	27.	2	8.	10.	2176.	701.
26	2	2	440.	3.	3	1.	2.	12.	0.
27	1	12	5754.	31.	2	9.	10.	2589.	288.
27	2	2	440.	3.	3	1.	2.	12.	0.
30	1	13	5760.	45.	1	24.	25.	3600.	143.
30	1	13	5760.	45.	2	8.	11.	3125.	824.
30	1	13	5760.	45.	3	25.	25.	3240.	0.
30	2	2	440.	4.	2	23.	25.	125.	11.
30	2	2	440.	4.	3	0.	3.	36.	0.
35	1	15	5772.	23.	1	20.	25.	3607.	727.
35	1	15	5772.	23.	2	6.	13.	3746.	2061.
35	1	15	5772.	23.	3	20.	25.	3247.	5.
35	2	2	440.	2.	2	20.	26.	127.	27.
35	2	2	440.	2.	3	-2.	5.	66.	1.
40	1	16	6619.	22.	1	23.	25.	3309.	257.
40	1	16	6619.	22.	3	17.	25.	3723.	8.
40	2	2	440.	2.	2	21.	26.	130.	28.
40	2	2	440.	2.	3	0.	8.	95.	1.
45	1	18	7612.	24.	1	18.	25.	3806.	1088.
45	1	18	7612.	24.	2	10.	17.	3280.	1282.
45	1	18	7612.	24.	3	17.	25.	4282.	10.
45	2	2	440.	2.	2	21.	27.	133.	28.
45	2	2	440.	2.	3	3.	10.	125.	1.
50	1	18	7612.	27.	1	20.	25.	3806.	761.
50	1	18	7612.	27.	2	12.	19.	3685.	1356.
50	1	18	7612.	27.	3	20.	25.	4282.	6.
50	2	2	440.	2.	2	22.	27.	136.	28.
50	2	2	440.	2.	3	5.	13.	154.	1.
55	1	18	7612.	26.	2	14.	21.	4089.	1356.
55	1	18	7612.	26.	3	20.	25.	4282.	6.
55	2	2	440.	2.	2	22.	28.	138.	28.
55	2	2	440.	2.	3	10.	15.	151.	0.
60	1	24	10972.	25.	1	17.	25.	4114.	1317.
60	1	24	10972.	25.	2	11.	24.	6478.	3382.
60	1	24	10972.	25.	3	14.	25.	6172.	20.

VIII-47

NUR RUN

NUR RUN

DAY	THR	UNITS	STRENGTH	REPLACEMENTS	TYPE	DAYS OH	DAYS DESRO	TONS OH	TONS REQD
60	2	2	440.	2.	2	23.	28.	141.	28.
60	2	2	440.	2.	3	10.	18.	175.	1.
65	1	24	10972.	33.	1	20.	25.	4114.	823.
65	1	24	10972.	33.	2	19.	26.	7062.	1955.
65	1	24	10972.	33.	3	20.	25.	6172.	9.
65	2	2	440.	2.	2	23.	29.	144.	28.
65	2	2	440.	2.	3	13.	20.	199.	1.
70	1	24	10972.	33.	1	20.	25.	4114.	823.
70	1	24	10972.	33.	2	21.	28.	7645.	1955.
70	1	24	10972.	33.	3	20.	25.	6172.	9.
70	2	2	440.	2.	2	24.	29.	147.	28.
70	2	2	440.	2.	3	15.	23.	223.	1.
END OF MDF2 FILE. DAY = 75									
75	1	25	11063.	33.	1	20.	25.	4149.	852.
75	1	25	11063.	33.	3	20.	25.	6223.	9.
75	2	2	440.	2.	2	24.	30.	149.	28.
75	2	2	440.	2.	3	18.	25.	247.	1.
80	1	25	11063.	33.	1	20.	25.	4149.	830.
80	1	25	11063.	33.	3	20.	25.	6223.	9.
80	2	2	440.	2.	2	25.	30.	149.	25.
80	2	2	440.	2.	3	20.	25.	247.	0.
85	1	25	11063.	33.	1	20.	25.	4149.	830.
85	1	25	11063.	33.	3	20.	25.	6223.	9.
85	2	2	440.	2.	2	25.	30.	149.	25.
85	2	2	440.	2.	3	20.	25.	247.	0.
90	1	25	11063.	33.	1	20.	25.	4149.	830.
90	1	25	11063.	33.	3	20.	25.	6223.	9.
90	2	2	440.	2.	2	25.	30.	149.	25.
90	2	2	440.	2.	3	20.	25.	247.	0.
UNITS IN THEATER 25 2 0 0 0 0 0 0 0 0									
CARDS PRODUCED 180									
FINAL SEQUENCE NUMBERS 255 52 0 0 0 0 0 0 0 0									

VIII-48

GPMD.F
PMD NOT ALLOWED

GPRT.1
FURPUR 27R1 RL71-3 08/24/76 19:01:03
UNCLASSIFIED*TPFS(0).F40/30.T
UNCLASSIFIED*32F26(1).F40/30.A 26,
UNCLASSIFIED*32F24(1).F40/30.A 24,
UNCLASSIFIED*32F23(1).F40/30.A 23,
UNCLASSIFIED*32F12(1).F40/30.A 12,
UNCLASSIFIED*58CAMP(1).F14/24.AX CSINTNAMES,
UNCLASSIFIED*XA(0).F40/30.T SA,
UNCLASSIFIED*XB(0).F14/24.T SB,

GED.R 32F24.
READ-ONLY MODE
CASE UPPER ASSUMED

NUR RUN

NUR RUN

EO 15-00-08/24/76-19:01:03-(0.)

EOIT

0027A1A	SGPWS DRY	PKGZ	999000	XPQXPPQKXPK000000
0027A1W	005000			
0028A1A	SAPWS AMMO	PKGZ	999000	XPQXPPQKXPK000000
0028A1W	000800			
0029A1A	P2PWS POL	POLZ	999000	XPQXPPQKXPK000000
0029A1W	000040			
0003A2A	SGPWS DRY	PKGZ	999000	WTGRWTGRWTGR000000
0003A2W	001000			
0004A2A	SAPWS AMMO	PKGZ	999000	WTGRWTGRWTGR000000
0004A2W	000200			
0005A2A	P2PWS POL	POLZ	999000	WTGRWTGRWTGR000000
0005A2W	000001			
0031A1A	SARESUPPLY AMMPKGP	010018		AKWQXPQKXPK000000
0031A1W	000943			
0049A1A	SARESUPPLY AMMPKGP	010019		AKWQXPQKXPK000000
0049A1W	000144			
0067A1A	SARESUPPLY AMMPKGP	010020		AKWQXPQKXPK000000
0067A1W	000583			
0006A2A	P2RESUPPLY POLPOLP	010020		HQLHWTGRWTGR000000
0006A2W	000000			
0085A1A	SARESUPPLY AMMPKGP	010021		AKWQXPQKXPK000000
0085A1W	000203			
0007A2A	P2RESUPPLY POLPOLP	010021		HQLHWTGRWTGR000000
0007A2W	000000			
0103A1A	SARESUPPLY AMMPKGP	010022		AKWQXPQKXPK000000
0103A1W	000203			
0008A2A	P2RESUPPLY POLPOLP	010022		HQLHWTGRWTGR000000
0008A2W	000000			
0121A1A	SARESUPPLY AMMPKGP	010023		AKWQXPQKXPK000000
0121A1W	000458			
0009A2A	P2RESUPPLY POLPOLP	010023		HQLHWTGRWTGR000000
0009A2W	000000			
0139A1A	SARESUPPLY AMMPKGP	010024		AKWQXPQKXPK000000
0139A1W	000245			
0010A2A	P2RESUPPLY POLPOLP	010024		HQLHWTGRWTGR000000
0010A2W	000000			
0157A1A	SARESUPPLY AMMPKGP	010025		AKWQXPQKXPK000000
0157A1W	000245			
0011A2A	P2RESUPPLY POLPOLP	010025		HQLHWTGRWTGR000000
0011A2W	000000			
0175A1A	SARESUPPLY AMMPKGP	010026		AKWQXPQKXPK000000
0175A1W	000701			
0012A2A	P2RESUPPLY POLPOLP	010026		HQLHWTGRWTGR000000
0012A2W	000000			
0193A1A	SARESUPPLY AMMPKGP	010027		AKWQXPQKXPK000000
0193A1W	000288			
0013A2A	P2RESUPPLY POLPOLP	010027		HQLHWTGRWTGR000000
0013A2W	000000			
0210A1A	SGRESUPPLY DRYPKGP	010026		RPCAXPQKXPK000000
0210A1W	000143			
0211A1A	SARESUPPLY AMMPKGP	010026		AKWQXPQKXPK000000
0211A1W	000824			
0212A1A	P2RESUPPLY POLPOLP	010026		HQLHWTGRWTGR000000
0212A1W	000000			
0213A1A	SGREPLACEMENTSPKGA	010026		HEKXPQKXPK0000045

VIII-49

NUR RUN

NUR RUN

001412A	SARESUPPLY AMMPKGP	010026	AKQWGTGRWTGR000000
001412W	000012		
001512A	P2RESUPPLY POLPOLP	010026	HQLHWTGRWTGR000000
001512W	000000		
001612A	SGREPLACEMENTSPKGA	010026	HEKPMTGRWTGR000004
001612W	SGRESUPPLY DRYPKGP	010031	RPCAXPQKXPQK000000
001712A	000727		
001712W	SARESUPPLY AMMPKGP	010031	AKWQXPQKXPQK000000
001812A	002061		
001812W	P2RESUPPLY POLPOLP	010031	HQLHXPQKXPQK000000
001912A	000005		
001912W	SGREPLACEMENTSPKGA	010031	HEKXPQKXPQK000023
002012A	SARESUPPLY AMMPKGP	010031	AKQWGTGRWTGR000000
002012W	000027		
002112A	P2RESUPPLY POLPOLP	010031	HQLHWTGRWTGR000000
002112W	000001		
002212A	SGREPLACEMENTSPKGA	010031	HEKPMTGRWTGR000002
002212W	SGRESUPPLY DRYPKGS	010036	RPCAXPQKXPQK000000
002312A	000257		
002312W	P2RESUPPLY POLPOLP	010036	HQLHXPQKXPQK000000
002412A	000008		
002412W	SGREPLACEMENTSPKGA	010036	HEKXPQKXPQK000022
002512A	SARESUPPLY AMMPKGS	010036	AKQWGTGRWTGR000000
002512W	000028		
002612A	P2RESUPPLY POLPOLP	010036	HQLHWTGRWTGR000000
002612W	000001		
002712A	SGREPLACEMENTSPKGA	010036	HEKPMTGRWTGR000002
002712W	SGRESUPPLY DRYPKGS	010041	RPCAXPQKXPQK000000
002812A	001088		
002812W	SARESUPPLY AMMPKGS	010041	AKWQXPQKXPQK000000
002912A	001282		
002912W	P2RESUPPLY POLPOLP	010041	HQLHXPQKXPQK000000
003012A	000010		
003012W	SGREPLACEMENTSPKGA	010041	HEKXPQKXPQK000024
003112A	SARESUPPLY AMMPKGS	010041	AKQWGTGRWTGR000000
003112W	000028		
003212A	P2RESUPPLY POLPOLP	010041	HQLHWTGRWTGR000000
003212W	000001		
003312A	SGREPLACEMENTSPKGA	010041	HEKPMTGRWTGR000002
003312W	SGRESUPPLY DRYPKGS	010046	RPCAXPQKXPQK000000
003412A	000761		
003412W	SARESUPPLY AMMPKGS	010046	AKWQXPQKXPQK000000
003512A	001356		
003512W	P2RESUPPLY POLPOLP	010046	HQLHXPQKXPQK000000
003612A	000006		
003612W	SGREPLACEMENTSPKGA	010046	HEKXPQKXPQK000027
003712A	SARESUPPLY AMMPKGS	010046	AKQWGTGRWTGR000000
003712W	000028		
003812A	P2RESUPPLY POLPOLP	010046	HQLHWTGRWTGR000000
003812W	000001		
003912A	SGREPLACEMENTSPKGA	010046	HEKPMTGRWTGR000002
003912W	SARESUPPLY AMMPKGS	010051	AKWQXPQKXPQK000000
004012A	001356		
004012W	P2RESUPPLY POLPOLP	010051	HQLHXPQKXPQK000000
004112A	000006		
004112W	SGREPLACEMENTSPKGA	010051	HEKXPQKXPQK000026
004212A	SARESUPPLY AMMPKGS	010051	AKQWGTGRWTGR000000

VIII-50

NUR RUN

NUR RUN

0029A2W	000028	P2RESUPPLY	POLPOL5	010051	HQLHWTGRWTCR0000000
0030A2A	000000	SGREPLACEMENTSPKGA	010051	HEKPMTGRWTCR0000002	
0030A2W	000000	SGREPLACEMENTSPKGA	010051	HEKPMTGRWTCR0000002	
0031A2A	001317	SGRESUPPLY	DRYPKGS	010056	RPCAXPQKXPK0000000
0032A1A	001317	SGRESUPPLY	DRYPKGS	010056	RPCAXPQKXPK0000000
0032A1W	001317	SGRESUPPLY	DRYPKGS	010056	RPCAXPQKXPK0000000
0033A1A	003342	SGRESUPPLY	DRYPKGS	010056	RPCAXPQKXPK0000000
0033A1W	003342	SGRESUPPLY	DRYPKGS	010056	RPCAXPQKXPK0000000
0034A1A	000020	SGRESUPPLY	DRYPKGS	010056	RPCAXPQKXPK0000000
0034A1W	000020	SGRESUPPLY	DRYPKGS	010056	RPCAXPQKXPK0000000
0035A1A	000028	SGRESUPPLY	DRYPKGS	010056	RPCAXPQKXPK0000000
0035A1W	000028	SGRESUPPLY	DRYPKGS	010056	RPCAXPQKXPK0000000
0036A1A	000028	SGRESUPPLY	DRYPKGS	010056	RPCAXPQKXPK0000000
0036A1W	000028	SGRESUPPLY	DRYPKGS	010056	RPCAXPQKXPK0000000
0037A1A	000028	SGRESUPPLY	DRYPKGS	010056	RPCAXPQKXPK0000000
0037A1W	000028	SGRESUPPLY	DRYPKGS	010056	RPCAXPQKXPK0000000
0038A1A	000028	SGRESUPPLY	DRYPKGS	010056	RPCAXPQKXPK0000000
0038A1W	000028	SGRESUPPLY	DRYPKGS	010056	RPCAXPQKXPK0000000
0039A1A	000028	SGRESUPPLY	DRYPKGS	010056	RPCAXPQKXPK0000000
0039A1W	000028	SGRESUPPLY	DRYPKGS	010056	RPCAXPQKXPK0000000
0040A1A	000028	SGRESUPPLY	DRYPKGS	010056	RPCAXPQKXPK0000000
0040A1W	000028	SGRESUPPLY	DRYPKGS	010056	RPCAXPQKXPK0000000
0041A1A	000028	SGRESUPPLY	DRYPKGS	010056	RPCAXPQKXPK0000000
0041A1W	000028	SGRESUPPLY	DRYPKGS	010056	RPCAXPQKXPK0000000
0042A1A	000028	SGRESUPPLY	DRYPKGS	010056	RPCAXPQKXPK0000000
0042A1W	000028	SGRESUPPLY	DRYPKGS	010056	RPCAXPQKXPK0000000
0043A1A	000028	SGRESUPPLY	DRYPKGS	010056	RPCAXPQKXPK0000000
0043A1W	000028	SGRESUPPLY	DRYPKGS	010056	RPCAXPQKXPK0000000
0044A1A	000028	SGRESUPPLY	DRYPKGS	010056	RPCAXPQKXPK0000000
0044A1W	000028	SGRESUPPLY	DRYPKGS	010056	RPCAXPQKXPK0000000
0045A1A	000028	SGRESUPPLY	DRYPKGS	010056	RPCAXPQKXPK0000000
0045A1W	000028	SGRESUPPLY	DRYPKGS	010056	RPCAXPQKXPK0000000
0046A1A	000028	SGRESUPPLY	DRYPKGS	010056	RPCAXPQKXPK0000000
0046A1W	000028	SGRESUPPLY	DRYPKGS	010056	RPCAXPQKXPK0000000
0047A1A	000028	SGRESUPPLY	DRYPKGS	010056	RPCAXPQKXPK0000000
0047A1W	000028	SGRESUPPLY	DRYPKGS	010056	RPCAXPQKXPK0000000
0048A1A	000028	SGRESUPPLY	DRYPKGS	010056	RPCAXPQKXPK0000000
0048A1W	000028	SGRESUPPLY	DRYPKGS	010056	RPCAXPQKXPK0000000
0049A1A	000028	SGRESUPPLY	DRYPKGS	010056	RPCAXPQKXPK0000000
0049A1W	000028	SGRESUPPLY	DRYPKGS	010056	RPCAXPQKXPK0000000
0050A1A	000028	SGRESUPPLY	DRYPKGS	010056	RPCAXPQKXPK0000000
0050A1W	000028	SGRESUPPLY	DRYPKGS	010056	RPCAXPQKXPK0000000

NUR RUN

004512A	P2RESUPPLY POLPOLS	010076	HQLHWTGRWTGR000000
004512W	000000		
004612A	SGREPLACEMENTSPKGA	010076	HEKPMTGRWTGR000002
025011A	SGRESUPPLY DRYPKGS	010081	RPCAXPKKXPQK000000
025011W	000830		
025111A	P2RESUPPLY POLPOLS	010081	HQLHXPQKXPQK000000
025111W	000009		
025211A	SGREPLACEMENTSPKGA	010081	HEKXPQKXPQK0000033
004712A	SARESUPPLY AMPPKGS	010081	AKQMTGRWTGR000000
004712W	000025		
004812A	P2RESUPPLY POLPOLS	010081	HQLHWTGRWTGR000000
004812W	000000		
004912A	SGREPLACEMENTSPKGA	010081	HEKPMTGRWTGR000002
025311A	SGRESUPPLY DRYPKGS	010086	RPCAXPKKXPQK000000
025311W	000830		
025411A	P2RESUPPLY POLPOLS	010086	HQLHXPQKXPQK000000
025411W	000009		
025511A	SGREPLACEMENTSPKGA	010086	HEKXPQKXPQK0000033
005012A	SARESUPPLY AMPPKGS	010086	AKQMTGRWTGR000000
005012W	000025		
005112A	P2RESUPPLY POLPOLS	010086	HQLHWTGRWTGR000000
005112W	000000		
005212A	SGREPLACEMENTSPKGA	010086	HEKPMTGRWTGR000002

EOF:192 SCAN:191

NO CORRECTIONS APPLIED.

*START START*24RTFT,MORSARUN

NUR RUN

START START.24RTEST.PKGRUN

Q RUN, /PRT A124F, H3883P3043K, UNCLASSIFIED, 10.30 . PKGRUN

Q HDG PACKAGE RUN

Q QUAL UNCLASSIFIED

Q ASG, AX *58CAMP

Q ASG, A 32F26. . MDF2 FILE

Q ASG, A 32F23. . NUR DATA (ORIG, DEST, RDD SEQUENCE)

Q ASG, A 32F22. . SMOBSMOD CARDS - OUTPUT (RDD SEQUENCE)

Q ASG, A 32F21. . PRINT OF UNITS IN PACKAGES

Q ASG, A 32F20. . PRINT OF ONE LINE/PACKAGE

Q ASG, A 32F19. . NUR PACKAGE DATA

Q ASG, A *24TOSV . OTHER SERVICE PACKAGE DATA

Q USE 26., 32F26.

Q USE 23., 32F23.

Q USE 22., 32F22.

Q USE 21., 32F21.

AD-A048 554

ARMY CONCEPTS ANALYSIS AGENCY BETHESDA MD
COMPUTER ASSISTED MATCH PROGRAM (CAMP), (U)
AUG 76 G L MARTIN, E R MONTAGNE
CAA-D-76-5

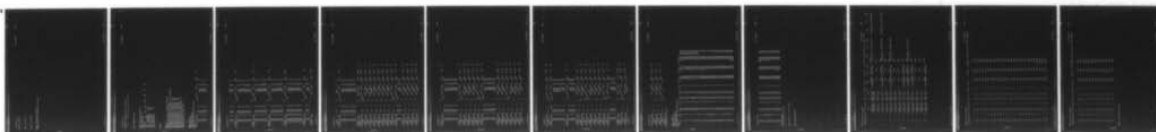
F/O 15/7

UNCLASSIFIED

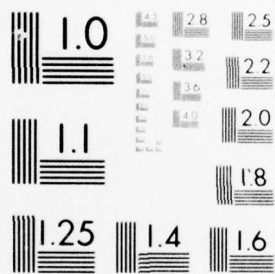
NL

4 OF 4

AD
A048554



END
DATE
FILMED
2-78
DDC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

PACKAGE RUN

DATE 082476

PAGE 3

0USE 20.132F20.

0USE 19.132F19.

0USE 18.124T0SV.

0PRT.5 24TESTDATA.PKGDATA
PURPUR 27RI RL71-3 08/24/76 19:03:09

VIII-55

PACKAGE RUN

DATE 082476

PAGE 3

PACKAGE RUN

UNCLASSIFIED*24TESTDATA(1).PKGDATA

1 10 1
2 1 999 4 999 7 999 8 999 9 999

QXQT *58CAMP*PKARS

QADD.P 24TESTDATA.PKGDATA

DELTA FOR RDD = 10

DELTA FOR AVL = 1

THEATER 1 4 7 8 9 9 9 0 0 0

DAY 999 999 999 999 999 999 999 999 999 999

UNITS PROCESSED 27

NUR PROCESSED 103

TOT # OF PKAGES 52

OF NUR PACKS 39

OTHER SERVICES 0

MAX UNITS IN PCK 5

GPMD.E

GPRT.1
FURPUR 27R1 RL71-3 08/24/76 19:03:24

UNCLASSIFIED*TPFS(0).F40/30.T
UNCLASSIFIED*58CAMP(1).F14/24.AX CSINTNAMES,

UNCLASSIFIED*32F24(1).F40/30.A 26,

UNCLASSIFIED*32F23(1).F40/30.A 23,

UNCLASSIFIED*32F22(1).F40/30.A 22,

UNCLASSIFIED*32F21(1).F40/30.A 21,

UNCLASSIFIED*32F20(1).F40/30.A 20,

UNCLASSIFIED*32F19(1).F40/30.A 19,

UNCLASSIFIED*24TOSV(1).F40/30.A 18,

UNCLASSIFIED*XA(0).F40/30.Y SA,

UNCLASSIFIED*XR(0).F2.T SB,

GED.R 32F22.

READ-ONLY MODE

CASE UPPER ASSUMED

ED 15.00-08/24/76-19:03:24-(0.)

EDJY

PACKAGE 000007

PACKAGE	000007	RDD=	2.00	AVL=	.00	1
.00	4 186 0	.780		1		
.00	4 186 0	.387	5	1		
.00	4 186 0	.004	7	1		
.00	4 186 0	.120	8	1		
.00	4 186 0	.365	9	1		
.00	4 186 0	.404	10	1		
.00	4 186 0	1.137	11	1		

PACKAGE RUN

DATE 082477

PACKAGE RUN									
.00	4	186	0			.073	13	1	
.00	4	186	0			.004	14	1	
.00	4	186	0			.214	15	1	
S444.									
PACKAGE 000012									
.00	4	227	0	.780	RDD=	2.00		AVL=	.00 2
.00	4	227	0			.387	5	2	
.00	4	227	0			.004	7	2	
.00	4	227	0			.120	8	2	
.00	4	227	0			.365	9	2	
.00	4	227	0			.404	10	2	
.00	4	227	0			1.137	11	2	
.00	4	227	0			.073	13	2	
.00	4	227	0			.004	14	2	
.00	4	227	0			.214	15	2	
S444.									
PACKAGE 000013									
.00	4	178	0	.863	RDD=	5.00		AVL=	.00 3
.00	4	178	0			1.760	5	3	
.00	4	178	0			.011	7	3	
.00	4	178	0			.636	10	3	
.00	4	178	0			.820	11	3	
.00	4	178	0			.004	12	3	
.00	4	178	0			.232	13	3	
.00	4	178	0			.006	14	3	
.00	4	178	0			.245	15	3	
S444.									
PACKAGE 000018									
.00	4	210	0	.851	RDD=	5.00		AVL=	.00 4
.00	4	210	0			1.760	5	4	
.00	4	210	0			.011	7	4	
.00	4	210	0			.617	10	4	
.00	4	210	0			.820	11	4	
.00	4	210	0			.004	12	4	
.00	4	210	0			.232	13	4	
.00	4	210	0			.006	14	4	
.00	4	210	0			.243	15	4	
S444.									
PACKAGE 000013									
.00	4	190	0	.440	RDD=	10.00		AVL=	10.00 5
.00	4	190	0			1.068	4	5	
.00	4	190	0			.025	5	5	
.00	4	190	0			.480	8	5	
.00	4	190	0			2.920	9	5	
.00	4	190	0			.295	10	5	
.00	4	190	0			.076	11	5	
.00	4	190	0			.001	12	5	
.00	4	190	0			.032	13	5	
.00	4	190	0			.001	14	5	
.00	4	190	0			.115	15	5	
S444.									
PACKAGE 000004									
.00	4	178	0	.006	RDD=	17.00		AVL=	.00 6
.00	4	178	0			.009	10	6	
.00	4	178	0			.001	15	6	
S444.									
PACKAGE 000014									
.00	4	178	0		RDD=	18.00		AVL=	10.00 7

PACKAGE RUN

5444.	0.	0.	0.	38.00	AVL=	.00	23
PACKAGE 000010							
.00	4	210	0	.993	1	23	
.00	4	210	0	1.962	5	23	
.00	4	210	0	.011	7	23	
.00	4	210	0	.365	9	23	
.00	4	210	0	.741	10	23	
.00	4	210	0	.856	11	23	
.00	4	210	0	.004	12	23	
.00	4	210	0	.236	13	23	
.00	4	210	0	.007	14	23	
.00	4	210	0	.303	15	23	
5444.	0.	0.	0.	0.			
PACKAGE 000026							
10.00	4	187	1	.077	1	-24	24
5444.	0.	0.	0.	0.			
PACKAGE 000037							
10.00	4	191	2	RDD= 41.00	AVL= 10.00	25	
5444.	0.	0.	0.	22.000	6	-25	
PACKAGE 000049							
10.00	4	216	2	RDD= 41.00	AVL= 10.00	26	
5444.	0.	0.	0.	1.849	16	-26	
PACKAGE 000021							
10.00	12	174	2	RDD= 41.00	AVL= 10.00	27	
5444.	0.	0.	0.	.084	17	-27	
PACKAGE 000031							
10.00	12	187	1	RDD= 41.00	AVL= 10.00	28	
5444.	0.	0.	0.	.006	1	-28	
PACKAGE 000006							
.00	4	179	0	RDD= 44.00	AVL= .00	29	
.00	4	178	0	.847	1	29	
.00	4	178	0	1.760	5	29	
.00	4	178	0	.011	7	29	
.00	4	178	0	.613	10	29	
.00	4	178	0	.820	11	29	
.00	4	178	0	.004	12	29	
.00	4	178	0	.232	13	29	
.00	4	178	0	.006	14	29	
.00	4	178	0	.242	15	29	
5444.	0.	0.	0.	0.			
PACKAGE 000017							
10.00	4	174	2	RDD= 46.00	AVL= 10.00	30	
5444.	0.	0.	0.	6.094	17	-30	
PACKAGE 000044							
10.00	12	191	2	RDD= 46.00	AVL= 10.00	31	
5444.	0.	0.	0.	2.000	6	-31	
PACKAGE 000027							
10.00	4	187	1	RDD= 56.00	AVL= 10.00	32	
5444.	0.	0.	0.	.091	1	-32	
PACKAGE 000038							
10.00	4	191	2	RDD= 56.00	AVL= 10.00	33	
5444.	0.	0.	0.	38.000	6	-33	
PACKAGE 000050							
10.00	4	216	2	RDD= 56.00	AVL= 10.00	34	
5444.	0.	0.	0.	2.963	16	-34	
PACKAGE 000022							
10.00	12	174	2	RDD= 56.00	AVL= 10.00	35	
5444.	0.	0.	0.	.084	17	-35	

VIII-59

PACKAGE RUN

PACKAGE RUN

PACKAGE 000032	10.00	12	187	1	.006	RDD=	56.00	1	AVL=	10.00	36
5444.					0.		0.		-36		
PACKAGE 000001	18003				3.253	RDD=	59.00	1	AVL=	.00	37
.00	4	210	1				4.100	5	37		
.00	4	210	1				2.400	8	37		
.00	4	210	1				6.205	9	37		
.00	4	210	1				2.287	10	37		
.00	4	210	1				3.439	11	37		
.00	4	210	1				.195	13	37		
.00	4	210	1				.016	14	37		
.00	4	210	1				.724	15	37		
5444.					0.		0.				
PACKAGE 000011					.107	RDD=	59.00	1	AVL=	.00	38
.00	4	210	0				.020	5	38		
.00	4	210	0				.032	10	38		
.00	4	210	0				.553	11	38		
.00	4	210	0				.025	15	38		
5444.					0.		0.				
PACKAGE 000018					0.	RDD=	61.00	1	AVL=	10.00	39
10.00	4	174	2		0.		3.910	17	-39		
5444.					0.		0.				
PACKAGE 000045					0.	RDD=	61.00	6	AVL=	10.00	40
10.00	12	191	2		0.		3.000	6	-40		
5444.					0.		0.				
PACKAGE 000028					0.	RDD=	71.00	1	AVL=	10.00	41
10.00	4	187	1		.099		0.	1	-41		
5444.					0.		0.				
PACKAGE 000039					0.	RDD=	71.00	6	AVL=	10.00	42
10.00	4	191	2		0.		27.000	6	-42		
5444.					0.		0.				
PACKAGE 000051					0.	RDD=	71.00	1	AVL=	10.00	43
10.00	4	216	2		0.		2.512	16	-43		
5444.					0.		0.				
PACKAGE 000023					0.	RDD=	71.00	17	AVL=	10.00	44
10.00	12	174	2		0.		.078	17	-44		
5444.					0.		0.				
PACKAGE 000033					0.	RDD=	71.00	1	AVL=	10.00	45
10.00	12	187	1		.006		0.	1	-45		
5444.					0.		0.				
PACKAGE 000002					0.	RDD=	75.00	1	AVL=	.00	46
.00	4	192	1		.091		.718	8	46		
.00	4	192	1		0.		1.460	9	46		
.00	4	192	1		0.		.002	13	46		
.00	4	192	1		0.		.004	15	46		
5444.					0.		0.				
PACKAGE 000046					0.	RDD=	76.00	1	AVL=	10.00	47
5444.					0.		0.				
PACKAGE 000029					0.	RDD=	86.00	1	AVL=	10.00	48
10.00	4	187	1		.033		0.	1	-48		
5444.					0.		0.				
PACKAGE 000040					0.	RDD=	86.00	6	AVL=	10.00	49
10.00	4	191	2		0.		9.000	6	-49		
5444.					0.		0.				

VIII-60

PACKAGE RUN

PACKAGE RUN

PACKAGE 000052 4 216 2 R00= 86.00 AVL= 10.00 50
 10.00 .830 16 -50
 5444.
 PACKAGE 000024 4 174 2 R00= 86.00 AVL= 10.00 51
 10.00 12 174 2 0. .025 17 -51
 5444.
 PACKAGE 000034 4 187 1 R00= 86.00 AVL= 10.00 52
 10.00 12 187 1 .002 0. 1 -52
 5444.

EOF:245 SCAN:244
 NO CORRECTIONS APPLIED.

8CD.R 32F20.
 READ-ONLY MODE
 CASE UPPER ASSUMED
 ED 15.00-08/24/76-19:03:26-(0.)

EDIT

PACK	DEST	ORIG	MODE	RDD	AVL	UNIT	TOT PAX	TOT STON	SQ FT/BRL
1	4	210	1	59.	0.	5	3.253	10.761	18003
2	4	192	1	75.	0.	1	.091	.006	8.605
3	4	178	0	5.	0.	4	.863	3.715	2.178
4	4	178	0	17.	0.	1	.006	.010	.000
5	4	178	0	29.	0.	3	.018	.031	.000
6	4	178	0	44.	0.	1	.847	3.689	.000
7	4	186	0	2.	0.	1	.780	2.223	.000
8	4	210	0	5.	0.	2	.851	3.694	.485
9	4	210	0	20.	0.	3	2.474	9.601	.000
10	4	210	0	38.	0.	2	.993	4.120	.485
11	4	210	0	59.	0.	1	.107	.630	.365
12	4	227	0	2.	0.	1	.780	2.223	.000
13	12	190	0	10.	10.	2	.440	.546	.485
14	4	174	0	18.	10.	11	.000	5.077	.000
15	4	174	0	31.	10.	1	.000	2.061	.000
16	4	174	2	31.	10.	1	.000	1.282	.000
17	4	174	2	46.	10.	3	.000	6.094	.000
18	4	174	2	61.	10.	2	.000	3.910	.000
19	12	174	0	26.	10.	2	.800	.039	.000
20	12	174	2	26.	10.	1	.000	.028	.000
21	12	174	2	41.	10.	3	.000	.084	.000
22	12	174	2	56.	10.	3	.000	.084	.000
23	12	174	2	71.	10.	3	.000	.078	.000
24	12	174	2	86.	10.	1	.000	.025	.000
25	4	187	1	26.	10.	3	.090	.000	.000
26	4	187	1	41.	10.	3	.077	.000	.000
27	4	187	1	56.	10.	3	.091	.000	.000
28	4	187	1	71.	10.	3	.099	.000	.000
29	4	187	1	86.	10.	1	.033	.000	.000
30	12	187	1	26.	10.	3	.008	.000	.000
31	12	187	1	41.	10.	3	.004	.000	.000
32	12	187	1	56.	10.	3	.004	.000	.000
33	12	187	1	71.	10.	3	.806	.000	.000
34	12	187	1	86.	10.	1	.802	.000	.000
35	4	191	0	26.	10.	2	.000	.000	.000
36	4	191	2	26.	10.	1	.000	.000	.000
37	4	191	2	41.	10.	3	.000	.000	.000

VIII-61

PACKAGE RUN

PACKAGE RUN

38	4	191	2	56.	10.	3	.000	.000	38.000	-25
39	4	191	2	71.	10.	3	.000	.000	27.000	-26
40	4	191	2	86.	10.	1	.000	.000	9.000	-27
41	12	191	0	20.	10.	9	.000	.000	.000	-28
42	12	191	0	31.	10.	1	.000	.000	1.000	-29
43	12	191	2	31.	10.	2	.000	.000	2.000	-30
44	12	191	2	46.	10.	3	.000	.000	2.000	-31
45	12	191	2	61.	10.	3	.000	.000	3.000	-32
46	12	191	2	76.	10.	3	.000	.000	.000	-33
47	4	216	0	26.	10.	2	.000	.870	.000	-34
48	4	216	2	26.	10.	1	.000	.257	.000	-35
49	4	216	2	41.	10.	2	.000	1.849	.000	-36
50	4	216	2	56.	10.	3	.000	2.963	.000	-37
51	4	216	2	71.	10.	3	.000	2.512	.000	-38
52	4	216	2	86.	10.	1	.000	.830	.000	-39

EOF:53 SCAN:52
NO CORRECTIONS APPLIED.

ENDG PRINT OF UNIT PACKAGES

QXQT 58CAMP.PRTPK

VIII-62

PRINT OF UNIT PACKAGES

DATE 082476 PAGE 11

PACKAGE DEST ORIG MODE RDO AVL UNITS TOT PAX TOT STON TOT SQ FT (1000'S OF PAX, TONS, SQ FT)

		SRC	UIC	TPSN	SRC	UIC	TPSN	SRC	UIC	TPSN	SRC	UIC	TPSN
1	4	210	1	59	0	5	3.253	10.761	8.605	17055H01000	TEST06 18003	17055H01000	TEST08 18003
		17387H50000	TEST09 18003										
		17052H00000	TEST05 18003										
2	4	192	1	75	0	1	.091	.006	2.178				
		05079H40000	TEST20 21435										
3	4	178	0	5	0	4	.863	3.715	.000	05510H2F801	X50003 31300	05510H2F801	X50004 31300
		05115H30000	TEST15 21364										
4	4	178	0	17	0	1	.006	.010	.000				
		05510H2F801	X50005 31300										
5	4	178	0	29	0	3	.018	.031	.000	05510H2F801	X50007 31300	05510H2F801	X50008 31300
		05510H2F801	X50006 31300										
6	4	178	0	44	0	1	.847	3.489	.000				
		05115H50000	X50001 31300										
7	4	186	0	2	0	1	.780	2.223	.485				
		05035H50000	TEST10 21353										
8	4	210	0	5	0	2	.851	3.494	.000	05510H2F800	TEST23 31557		
		05115H30000	TEST16 21364										
9	4	210	0	20	0	3	2.474	9.601	.485	05115H30000	TEST17 21364	05115H30000	TEST18 21364
		05035H50000	TEST13 21353										
10	4	210	0	38	0	2	.993	4.120	.365				
		05115H30000	TEST19 21364										
11	4	210	0	59	0	1	.107	.630	.000	05327G60000	TEST21 31531		
		05124G60000	TEST24 31436										
12	4	227	0	2	0	1	.780	2.223	.485				
		05035H50000	TEST11 21353										
13	12	190	0	10	10	2	.440	.546	4.468	01207H20000	TEST04 30612		
		01127H10000	TEST03 30623										

VIII-63

PRINT OF UNIT PACKAGES

DATE 082476

PAGE

12

PACKAGE DEST ORIG MODE RCD AVL ENTRY TOT PAX TOT STON TOT BARRL (1000'S OF PAX, TONS, BARRELS)

14	4	174	0	18	10	11	.000	5.077	.000
15	4	174	0	31	10	1	.000	2.061	.000
16	4	174	2	31	10	1	.000	1.282	.000
17	4	174	2	46	10	3	.000	6.094	.000
18	4	174	2	61	10	2	.000	3.910	.000
19	12	174	0	26	10	2	.000	.039	.000
20	12	174	2	26	10	1	.000	.028	.000
21	12	174	2	41	10	3	.000	.084	.000
22	12	174	2	56	10	3	.000	.084	.000
23	12	174	2	71	10	3	.000	.078	.000
24	12	174	2	86	10	1	.000	.025	.000
25	4	187	1	26	10	3	.090	.000	.000
26	4	187	1	41	10	3	.077	.000	.000
27	4	187	1	56	10	3	.091	.000	.000
28	4	187	1	71	10	3	.099	.000	.000
29	4	187	1	86	10	1	.033	.000	.000
30	12	187	1	26	10	3	.008	.000	.000
31	12	187	1	41	10	3	.006	.000	.000
32	12	187	1	56	10	3	.006	.000	.000
33	12	187	1	71	10	3	.006	.000	.000
34	12	187	1	86	10	1	.002	.000	.000
35	4	191	0	26	10	2	.000	.000	5.000
36	4	191	2	26	10	1	.000	.000	8.000
37	4	191	2	41	10	3	.000	.000	22.000
38	4	191	2	56	10	3	.000	.000	38.000
39	4	191	2	71	10	3	.000	.000	27.000

VIII-64

PRINT OF UNIT PACKAGES

DATE 082476

PAGE

12

PRINT OF UNIT PACKAGES

DATE 082476 PAGE 13

PACKAGE DEST ORIG MODE RDD AVL ENTRY TOT PAX TOT STON TOT BARRL (1000'S OF PAX. TONS. BARRELS

40	4	191	2	86	10	1	.000	.000	9.000
41	12	191	0	20	10	9	.000	.000	.000
42	12	191	0	31	10	1	.000	.000	1.000
43	12	191	2	31	10	2	.000	.000	2.000
44	12	191	2	46	10	3	.000	.000	2.000
45	12	191	2	61	10	3	.000	.000	3.000
46	12	191	2	76	10	3	.000	.000	.000
47	4	216	0	26	10	2	.000	.870	.000
48	4	216	2	26	10	1	.000	.257	.000
49	4	216	2	41	10	2	.000	1.849	.000
50	4	216	2	56	10	3	.000	2.963	.000
51	4	216	2	71	10	3	.000	2.512	.000
52	4	216	2	86	10	1	.000	.830	.000

VIII-65

MSG.N START024R.LAYINRUN

PRINT OF UNIT PACKAGES

DATE 082476 PAGE 13

78